NASA Contract Report CR-182236

The MHOST Finite Element Program:

3-D Inelastic Analysis Methods for Hot Section Components

Volume III - Systems' Manual

Shohei Nakazawa
The Finite Element Factory
Sunnyvale, California

July 1989

Prepared for
Lewis Research Center
Under Contract NAS3 - 23697

National Aeronautics and
Space Administration

1. Report No. CR-182236	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle The MHOST Finite Element Program: 3-D Inelastic Analysis Methods for Hot Section Components Volume III - Systems' Manual		5. Report Date July, 1989
		6. Performing Organization Code 533-04-11
7. Author(s) Shohei Nakazawa		8. Performing Organization Report No. FEF/PD/1102/89
		10. Work Unit No.
9. Performing Organization Name and Address MARC Analysis Research Corporation, 260 Sheridan Avenue, Palo Alto, CA 94306 The Finite Element Factory, 1767 Wright Avenue, Sunnyvale, CA 94087		11. Contract or Grant No. NAS3-23697
		13. Type of Report and Period Covered Final Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration, Lewis Research Center Project Manager, D. A. Hopkins, MS49-8 21000 Brookpark Road, Cleveland, OH 44135		14. Sponsoring Agency Code RTOP 533-04-1A

15. Supplementary Notes

CR-182205 The MHOST Finite Element Program Volume I - Theoretical Manual

CR-182235 The MHOST Finite Element Program Volume II - Users Manual

16. Abstract

This document discusses the internal structure of the MHOST finite element program designed for three-dimensional inelastic analysis of gas turbine hot section components. This computer code is the first implementation of the mixed iterative solution strategy for improved efficiency and accuracy over the conventional finite element method.

This document covers the control structure of the program, the data storagescheme and the memory allocation procedure and the file handling facilities including the read/write sequences.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
3-D Inelastic Analysis, Finite Elements, High Temperature, Creep, Vibration, Buckling, Solution Methods, Constitutive Modeling	General Release

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified		

*For sale by the National Technical Information Service, Springfield, Virginia 22161

MHOST Version 4.2 Systems Manual

This document discusses the internal structure of the MHOST finite element program designed for three-dimensional inelastic analysis of gas turbine engine hot section components. The computer code is the first implementation of the mixed iterative solution strategy for improved efficiency and accuracy over the conventional finite element method. The formulation of the mixed iterative solution method is an original development under the HOST project (contract NAS3-23697) and detailed documaentation is available in the MHOST Theoretical Manual and other publications. The computer program has been written, tested and maintained at MARC Analysis Research Corporation, Advanced Project Group as a subcontractor to the United Technology Pratt and Whitney Aircraft.

The complete computer program consists of about 450 subroutines and a total of 47,000 lines of Fortran 77 statements. The current version 4.2 is no longer compatible with theANSI Fortran 66 standard.

This document covers:

(i) the control structure of the program;

(ii) the data storage scheme and the memory allocation procedure;

(iii) the file handling facilities including the read/write sequences.

A brief note on the control variables in the labelled common blocks is given in the control structure section. Pointers for the working arrays in the common block are described in the second section of this document. The files used internally and those produced to communicate with other softwares (such as the graphic post-processing systems) are discussed in detail in the last section.

The appandix includes the brief description of each subroutine in conjunction with the names of common block referenced therein.

The MHOST code has been developed on PRIME 9955 at MARC running under Primos operationg system (Rev.19.4.2 of F77 compiler has been used) and recently ported over to Alliant FX/8 running under unix operationg system. This version is protable to any other unix-based computers with minimum amount of the conversion work. Versions are available on IBM mainframes with VS-Fortran Compiler (tested at United Technology Pratt and Whitney Aircraft) and CRAY X-MP COS using CFT compiler (the installation at NASA Lewis Research Center).

TABEL OF CONTENTS

0.0 INTRODUCTION 1

0.1 Program Architecture 1

0.2 Rules and Conventions 3

0.3 Subroutines and Common Blocks 5

0.4 System Dependency 10

1.0 CONTROL STRUCTURE 17

1.1 Overview 17

1.2 Execution Supervisor and Analysis Drivers 22

1.3 Control Variables 29

2.0 DATA STORAGE SCHEME 39

2.1 Overview 39

2.2 Memory Allocation Subprogram 40

2.3 Counters and Pointers 96

3.0 THE FILE SYSTEM 104

3.1 Overview 104

3.2 User Interface 105

3.3 Restart File 136

3.4 Post-Processing Data File 149

APPENDIX Subroutines A-1

0.0 INTRODUCTION

The MHOST program has evolved from a small finite element code for testing new and innovative ideas into a versatile package usable as a research and development tool in solid and structural mechanics. Several code developers have worked on this program at various times over four years; more than 47,000 lines of extensively commented Fortran 77 code now make up the program. This document describes the internal architecture of the code. Particular emphasis is placed here on the facilities by which users can modify the code to incorporate new ideas such as different elements or constitutive equations.

The concept of three libraries plays a central role in the MHOST code. These libraries are:

The element library. Almost all the element specific operations are coded in this library with a common interface subroutine. Note that the MHOST code is written based on the mixed iterative solution concept and the element library is accessed significantly more often have than in the case for conventional displacement method codes.

The material library. All the constitutive equations built into the MHOST program are accessed through an interface routine for the material library. The nodal evaluation of the constitutive equations in the mixed iterative solution process enables this operation to occur independently of the loop structure for the element formulation.

The solution algorithm library. The MHOST code uses a number of modern iterative solution algorithms and two types of solvers for a linearized system of algebraic equations. These options are accessed directly from the finite element driver routines; from a programming point of view, the routines in the solution algorithm library are not as clearly identifiable as other library routines.

0.1 Program Architecture

The MHOST program is designed and constructed to perform inelastic finite element computations in a reliable manner. A number of analysis driver routines are coded to perform individual clearly defined tasks. Inter-relations between analysis tasks are controlled by the execution supervisor subprogram and the utilities attached to it. All three libraries discussed briefly in the previous section are executed by each analysis driver routine. The program architecture of the MHOST program is illustrated In Figure 1.

```
                         ------------------------
                         | Execution Supervisor |
                         |   SUBROUTINE MHOST   |
                         ------------------------
                                    |
             -----------------------|------------------------
             |                      |                       |
 -------------------------          |         -------------------------
 | Data input Subsystem  |          |         | Bureaucrats           |
 | SUBROUTINE DATIN1 and |          |         | SUBROUTINE LETCMD and |
 | SUBROUTINE BULKIN     |          |         | SUBROUTINE RUNCMD     |
 -------------------------          |         -------------------------
                   -----------------------------------
                   | Finite Element Analysis Modules |
                   -----------------------------------
                                    |
             -----------------------|------------------------
             |                      |                       |
 ------------------------------     |     ----------------------------------
 | Incremental Data Input and |     |     |  Incremental Report Generation |
 | Array Intializations       |     |     ----------------------------------
 ------------------------------     |
                     -------------------------------
                     | Element Assembly Submodules |
                     -------------------------------
                                    |
             -----------------------|------------------------
             |                      |                       |
 -------------------      ----------------------    -----------------
 | Element Library |      | Constitutive Model |    | Utilities for |
 -------------------      | Library            |    | Linear Algebra|
                          ----------------------    -----------------
```

Figure 1 Architecture of MHOST Program Version 4.2

The concept of multiple-drivers prevents users from combining unreasonable options, and avoids program troubles. Also this coding strategy allows code developers to add new analysis capabilities without affecting existing functions of the MHOST program.

The nonlinear finite element computations are arranged in a nested loop structure. The outer loops which control the algorithmic operations are coded in the analysis driver routines, while the inner loops involving incremental data addressing are coded in the element assembly submodules, a level below the analysis drivers.

A unique feature of MHOST is that all information is stored at nodes. The finite element nodal database resides in the lower address of working space in the blank common block. The storage allocation for this nodal database is virtually independent of analysis and solution algorithms and takes place before reading the finite element model data.

The working space required to perform the solution is allocated in the higher address portion of the same working space. The allocation of this working space is analysis dependent and takes place inside of the finite element driver routine.

0.2 Rules and Conventions

Coding rules are established to maintain the readability and maintainability of the MHOST program.

Subroutine names are selected as close as possible to a plain English word representing the operation performed by the subroutine. Because of limitations set by the old Fortran standard of six characters, some subroutine names can be cryptic.

A special rule has been implemented for subroutines in the element library:

SnnmmN Shape functions for elements consisting of mm-nodal points in nn-geometrical configuration.

DnnmmN Cartesian derivatives for elements consisting of mm-nodal points in nn-geometrical configuration.

nn	Geometry
2D	Two dimensional
AX	Axi-symmetric

3D	Three dimensional
SH	Three dimensional shell

Some of the other rules are:

- The subroutine names with string IN are used for data read from the main input channel.
- The string INIT is used to indicate subprograms involving memory allocation, except for INITST which generates the initial stress terms in the stiffness equations.
- The string SUB appears in subroutine names for subroutines involving the subelement scheme and the local-global analysis.
- The subroutine names starting with U indicate user subroutines.

Variable names are chosen as close as possible to a plain English word representing the information contained in the variables. Because of limitations set by the old Fortran standard of six characters, some variable names can be often cryptic.

Note that the implicit data types of Fortran specification are assumed. Variable names starting with I, J, K, L, M and N are integers except for those specifically declared as logical variables.

Some of the variable name conventions are:

- The string MAX is used for the maximum number of entries in adjustable dimensioned arrays.
- The string NEL is used for the actual number of entries in adjustable dimensioned arrays containing element related information.

Statement Numbers

Four-digit statement numbers are used for better visibility of the Fortran program listing. Codes which are ported from other software do not necessarily follow the conventions defined below.

The format statements referred from the read statements are labelled by statement numbers from 1000 to 1999. The format statements referred from the write statements are labelled by statement numbers from 2000 to 2999.

Ends of DO loops are labelled by statement numbers from 5000 to 5999. Statements to which control is transferred by the conditional GO TO statements are labelled by statement numbers from 8000 to 8999.

Efforts have been made to avoid arithmetic IF and GO TO statements in the MHOST code. Instead the block IF structure is implemented wherever possible. Note that this approach has made the code incompatible with Fortran 66 compilers.

0.3 Subroutines and Common Blocks

All the subroutines included in the MHOST program Version 4.2 are listed below in alphabetical order:

ACCLIN	ADAPTC	ADAPTD	ADAPTS
ADD	ADDBAN	ADDIAG	ADDINC
ADDINV	ADDPUL	ADDSMU	ANPLAS
ASMVEC	ASSEM1	ASSEM2	ASSEM3
ASSEM4	ASSEM5	ATTRIB	BACSUB
BANDBR	BANNER	BASAXS	BASEIN
BASPSN	BASPST	BASSOL	BAXSYM
BEAMIN	BFGSLH	BFGSRH	BFGSVW
BFLOAD	BLCK01	BMSTRS	BNDTRM
BODYIN	BOUND1	BOUND2	BOUND3
BOUNDN	BOUNFR	BOUNIN	BPSTRN
BPSTRS	BREAD	BSHDKQ	BSHELL
BSOLID	BTBEAM	BUCKLE	BULKIN
BWRITE	CENMAS	CENT2D	CENT3D
CENTAX	CENTBM	CENTSH	CHARIN
CHCHAR	CHKELM	CLENUP	CLOSEF
CNODEL	CNSMAS	CNSTNM	COLRED
COMPDF	COMPIN	COMPRO	CONDSE
CONJUG	CONNIN	CONTIN	COORIN
COPY	COPYCH	COPYDS	COPYIN
COPYSD	CORDTR	COROUT	CPXBK1
CPXBK2	CPXDIV	CPXEXC	CPXFAC
CPXFOR	CPXMUL	CPXREA	CPXRES
CRPLAW	CRPSTN	CUTHIL	D2D04N
D2D09N	D3D08N	D3D27N	DAMPIN
DASHIN	DAT1	DAT2	DAT3
DAT5	DATBMS	DATCG1	DATER
DATIN1	DATIN2	DATIN3	DATOH1
DATOU1	DATOU4	DAX04N	DAX09N
DBM02N	DECINT	DECOMP	DECREA

DEFGUP	DERIV	DIAM	DIRECT
DISPIN	DISTIN	DIV2Q2	DIV2X2
DMATIN	DMPING	DOT	DSH04N
DSHELL	DSLOAD	DUPLIN	DXOUT
DYNAMC	DYNAMT	DYNOP	EIGENV
ELVULV	EQVC	EQVS	EQVSTR
ERROR	ETRANS	EXTEIN	FESOLN
FILINT	FILL	FIRST1	FIXINT
FOLOIN	FORRES	FREDOM	FREFOR
FRNTBL	FRNTOP	FRONTB	FRONTF
FRONTR	FRONTS	FRONTW	GAUSSP
GBM02N	GENCOR	GENNOD	GEOMAT
GETBSP	GO2GLO	GO2ROT	GRAV2D
GRAV3D	GRAVAX	GRAVBM	GRAVSH
GSH04N	HARMIN	HEAD	HOLECR
HOLEDF	HOLEIN	HOLELM	HOOKBM
HOOKLW	HOST	ICLEAR	INCRIN
INCSBC	INIMOP	INITDF	INITFR
INITI1	INITI2	INITIN	INITSE
INITST	INRDIR	INRFRC	INSIDE
INTDYN	INTERP	INTINT	INTSQQ
INV3	INVERT	ITERIN	JACOBI
JT	KEY	L2NORM	LAXSYM
LAYINT	LELAST	LETCMD	LEVEL
LINES	LINES2	LINESR	LINESU
LMPMAS	LOCVEC	LPSTRN	LPSTRS
LSHELL	LSOLID	LTBEAM	MADD
MAIN	MASMAT	MASSIN	MATINV
MATONE	MATPRT	MATSUM	MAXCON
MAXIM	MESURE	MID	MIDDLE
MKFAKE	MODAL	MULT	MULTT
NEWACC	NEWADD	NEWMRK	NEWRHS
NEWVEL	NODPRE	NODSTR	NOTION
NRMNRM	NUL	NULINT	NULNRM
OPTEST	OPTIM	ORIENT	OUTPRO
PAGE	PAGE2	PAGE2S	PAGE3
PAGE3S	PERDIN	PERDOP	PJOOP
PLASTD	PLASTS	PNTTNM	POLD2D
POLD3D	POLICE	POSTEN	POSTOU
POSTPR	PREFIN	PRELEM	PRESET
PRESIN	PRFRNT	PRINCV	PRININ
PRINO1	PRINO2	PRINO3	PRINOU
PRINSU	PRINTM	PRINTS	PRNSHL
PRNTEL	PRNTNO	PRTERR	PRWARN
PSDIN	PULSIN	PUTDUP	PUTTIE

QNODEL	QPSTRS	QUIT	R3DTEN
RAMDSK	RATIO	RBF	READEX
REASPR	REDIAG	RELDFG	RELOAD
RESCHK	RESCON	RESDYN	RESEQ1
RESEQ2	RESID	RESOLV	RESTRT
ROTBAK	ROTDMT	ROTFOR	ROTPRF
ROTTEN	ROW	RSHELL	RTBEAM
RTFOLF	RUNCMD	S1D02N	S2D04N
S2D09N	S3D08N	S3D27N	SAVER
SBCIN	SCALER	SEARCH	SEARCI
SELECT	SETCCM	SETHFN	SETOLR
SETOMD	SETRMS	SETUP	SHIFIN
SHOHEI	SHTRAN	SIMPLE	SK
SMASTR	SMULT	SNODEL	SOLUT1
SOLUT2	SOLUTN	SOLVER	SOLVIT
SPRIIN	SPSTRS	SSOLID	STATIC
STCKIO	STIFF	STRAIN	STRESS
STRING	STRIPB	STRSBC	STRUCT
SUBALC	SUBCHK	SUBDER	SUBDIV
SUBELV	SUBFEM	SUBGLB	SUBGLD
SUBINC	SUBINT	SUBRES	SUBSIN
SUBSO1	SUBSO2	SUBSOL	SUBSPC
SUBST1	SUBSTN	SUBSTS	SUBT
SUBVAL	SUPER	SYSEQN	T2D04N
T2D04P	T3D08N	TBM02N	TEMPIN
TFULL2	THRSTN	TIMEIN	TIMER
TIMOUT	TMULT	TMULTV	TNSPRD
TRACIN	TRANIN	TRANS1	TRANS2
TRANSP	TSH04N	TSHIFT	TYING1
TYING2	TYING3	TYININ	TYPEIN
UBOUN	UCOEF	UCOOR	UDERIV
UFORCE	UFXORD	UHOOK	UNITST
UPRESS	UPTX	UPTXL	USXX
UTEMP	UTHERM	VALINT	VDSKIO
VELCIN	VMULT	VSH04N	VSWELL
VTMULT	VULVRG	VVMULT	WALCON
WALKEQ	WKSLP	WORKIN	WRITEX
YIEL	YIELIN		

All the common blocks included in the MHOST program Version 4.2 are listed below in alphabetical order:

```
      COMMON / ADDVAL / ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
```

```
2                       PI     ,LINE   ,LINE2
 COMMON / AUTOIN / CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
1                  JADAP ,NCREEP,SCALE
 COMMON / BODYFR / POINTS(      3,       2)
 COMMON / BSECT  / IBSECT,KBSECT
 COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
+                  JWKSLP,JPRES ,JCDUM2,JCDUM3
 COMMON / COMPND / NXSTAT,NXSOLV,NXINTG,NXMODL,NXBCKL,NXSUPR,
1                  NXREQN,NXDUM1,NXDUM2,NXDUM3,NXDUM4
 COMMON / COUNT  / LININC,LINTOT,NOECHO
 COMMON / CTITLE / TITLE (    20),IDAT  (     5),IDATE2,ICLOCK,
1                  IFCRAY
 COMMON / DAMP   / DAMPF(3)
 COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
 COMMON / MODSUP / IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
 COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
 COMMON / ELTYP  / NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELPR ,
1                  NELINT,NELLV ,NELLAY,NDI   ,NSHEAR,NELCMP
 COMMON / ERRORS / IERR
 COMMON / FREE   / IA    (    80),IBEGIN(    16),ILENGT(    16),
1                  NSTRIN,IS    ,ICOL  ,NEW
 LOGICAL                                NEW
 COMMON / HARMON / OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
1                  ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
2                  ICCMAT
 COMMON / INCCON / FACTOR,INCFLG(21)
 COMMON / MACHIN / IDP
 COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
 COMMON / LOUBIN / JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
1                  JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
```

```
      COMMON / PAGCNT / NPAGE1,NPAGE2
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / PERPAR / IPTYPE(    32),NPTYPE,NPVARS,NPSETS,JPERT ,       NESSUS
     1                  NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012         NESSUS
      COMMON / PERPTR / IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,        NESSUS
     1                  IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,        NESSUS
     2                  IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK         NESSUS
      COMMON / PERDAT / IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,        NESSUS
     1                  IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,        NESSUS
     2                  IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND         NESSUS
      COMMON / PERIOD / JPEROD(    2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / POSTPN / IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(  6   )
      COMMON / POWER  / IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                  IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                  IPSF  ,IPSD
      COMMON / PREPAR / NRFPTS,NRFSDS,NRFLIN,NRFINT,NRFREA,NRFSEL,        NESSUS
     1                  NRFCMP,NRFDOF,NRFSIZ,LRFEND,JRFTYP,JCOUNT,        NESSUS
     2                  JCFLAG,JRFSEL,JRFDOM,NVKEPT,MORE                  NESSUS
      COMMON / PREPTR / IRFPTS,IRFINT,IRFMNR,IRFSDR,IRFSEL,IRFSET,        NESSUS
     1                  IRFMNV,IRFSDV,IRFPTV,IRFVEC,IRFVAL,IRFNRM,        NESSUS
     2                  IRFCOR,IRFID ,IRFWRK                              NESSUS
      COMMON / PULSES / IPULSE,KPULSE,IPDTIM,IPDFOR
      COMMON / RESULT / MANVAR(    7 ),JPR   ,ICOM1 ,NCOMP
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                  IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                  ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                  IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                  IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                  IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                  IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                  IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                  IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                  IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
```

```
1                      IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
2                      IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
3                      IMASDI,IMASUP,IST521,IST522,IST523,IST524
 COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
1                      IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
2                      IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
3                      KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
4                      KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
5                      IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
6                      KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
7                      IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
8                      IST649,IST650,IST651,IST652,IST653,IST654
 COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
 COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
1                      IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
2                      KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
 COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
1                      KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
 COMMON / SUBELM / ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
 COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
1                      NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
 COMMON / SUBSTR / NLVSUB(    10),NFRSUB(    10)
 COMMON / SHIFT  / ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
 COMMON / TIME   / TIMINC,TOTINC,RUNTIM
 COMMON / TIMLOC / CPU0
 COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
 COMMON / TRANSF / CTRANS(     9),XJACOB(     9)
 COMMON / TOLER  / RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
1                      ENGTOR,ENGNRM
 COMMON / VRTDSK / ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
1                      IVPAGE,NVPAGE,IVSTRT,IKRO  ,ILCOL ,IPIVCO,
2                      IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND ,
3                      ISRECD,IERECD
 COMMON / ZPRINT / MAXCOL,NELCOL,LAYPRT,JTENSR
```

Most of the variables and arrays in the common blocks are used to store control variables and counters. The exceptions are START1 to START8 which contain all the pointer values for working storage allocation.

0.4 System Dependency

The MHOST Version 4.2 program is written in Fortran 77, as closely as possible to ANSI specification. The code has been developed on the PRIME 9955 under PRIMOS F77 Rev. 19.4.2.

In the original version, integer and single precision real word lengths of 32 bits are assumed and double precision is specified by using

IMPLICIT REAL*8 (A-H, O-Z)

for all real variables except for working storage arrays. The ratio of integer and double precision word length is stored in the variable IDP in common block MACHIN. The value is set at 2 in the main program of the original version.

To generate versions for 64 bits/word machines such as CRAYs and CYBERs, the value of IDP must be set 1 (one) in the main program.

The file opening and error handling procedures are system dependent. The subroutine INTINT is called from the main program to perform these functions. An example of this routine for the PRIME version is included here.

```
C=SUBROUTINE=INTINT  CALLED FROM THE MAIN PROGRAM
      SUBROUTINE INTINT
C
C ***********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C ***********************************************************************
C
      COMMON / VRTDSK / ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
     1                  IVPAGE,NVPAGE,IVSTRT,IKRO   ,ILCOL  ,IPIVCO,
     2                  IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND  ,
     3                  ISRECD,IERECD
C
      CHARACTER*20  NAMEIN , NAMEPR , NAMEME , LOGFIL , NAMESC ,
     1              NAME08 , NAME09
C
C --- DEFINE THE BUFFER SIZE FOR SCRATCH FILE -------------------------
C
      MAVAIL =   3 000
      NBUFFR =  MAVAIL * 4 + 12
C
C ***********************************************************************
C
C     OPEN FILES : NAMES READ FROM A SPECIAL FILE 'T$NAMES'
C
```

```
      OPEN(  UNIT = 18 , FILE = 'T$FILES' , FORM = 'FORMATTED ' )
C
      READ(  18 , 1000 ) NAMEIN
      READ(  18 , 1000 ) NAMEPR
      READ(  18 , 1000 ) NAMEME
      READ(  18 , 1000 ) LOGFIL
      READ(  18 , 1000 ) NAMESC
      READ(  18 , 1000 ) NAME08
      READ(  18 , 1000 ) NAME09
C
      OPEN(   5 , FILE =  NAMEIN  )
      OPEN(   6 , FILE =  NAMEPR  )
      OPEN(  19 , FILE =  NAMEME  )
C
      OPEN(   8 ,                    FORM = 'UNFORMATTED'  ,
     1            RECL  =  256     , FILE = NAME08         )
      OPEN(   9 ,                    FILE = NAME09         )
      OPEN(  12 ,                    FORM = 'UNFORMATTED'  ,
     1            RECL   =  NBUFFR , FILE = NAMESC         )
C
      CLOSE( 18 , STATUS = 'DELETE' )
C
C ***********************************************************************
C
 1000 FORMAT( A20 )
C
C ***********************************************************************
C
C
      R  E  T  U  R  N
      END
```

On the PRIME a user will have to prepare a file called T$FILES in the working directory containing names of all the files to be opened as Fortran I/O units.

On other systems, this operation can be performed either by issuing commands at the operating system level or by duplicating the PRIME version approach directly in the MHOST program.

Note it is recommended that on IBM machines the error and warning message print-outs be disabled for double precision floating point operations. This is due to the fact that the solution of an algebraic system of equations involves double precision variables with fictitiously large exponents.

The real time clock and calendar routine and the CPU clock routine both involve system calls and differ depending on the machinery and operating system. The organization of common block CTITLE in which the date and time are stored may need to be modified depending on how the system returns these values. The actual call to the system's clock and calendar is made through SUBROUTINE DATER. The following code is used to communicate with the PRIME/PRIMOS:

```
C  SUBROUTINE DATER
      SUBROUTINE DATER(IDAT,IDAT4)
C * * * * * * * * * *
C
C  DATE ROUTINE FOR THE PRIME
C
C * * * * * * * * * *
      INTEGER*2 ARRAYS(15),NUM,IDAT(6)
      DIMENSION IDAT4(2)
      NUM=4
      CALL TIMDAT(ARRAYS,NUM)
      IDAT(1)=ARRAYS(1)
      IDAT(2)=ARRAYS(1)
      IDAT(3)=ARRAYS(2)
      IDAT(4)=ARRAYS(2)
      IDAT(5)=ARRAYS(3)
      IDAT(6)=ARRAYS(3)
      I0=ARRAYS(4)
      I1=I0/60
      I2=I0-60*I1
      IDAT4(1)=I1
      IDAT4(2)=I2
      RETURN
      END
```

On the CRAY, this operation is simplified to two system calls in SUBROUTINE HEAD as shown below:

```
C=SUBROUTINE= HEAD    CALLED BY 'DATIN1'
      SUBROUTINE HEAD
     1    (VERNO ,MONTH ,JDATE ,ILPRNT,ICONSL)
C
C **********************************************************************
C
C    PICK UP TODAY'S DATE AND TIME AND THEN PRINT 'HOST' LOGO ON THE
C    FIRST PAGE OF LISTING. NOTE THIS OCCURS EVEN WHEN VACANT DATA FILE
C    IS ASSIGNED ( ON PRIME ).
```

```
C
C ***********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
C ***
      COMMON / CTITLE / TITLE (    20),IDAT  (     5),IDATE2,ICLOCK,
     1                  IFCRAY
C
C ***********************************************************************
C
      CALL CLOCK  ( ICLOCK )
      CALL DATE   ( IDATE2 )
C
      WRITE(ILPRNT,1000)
      WRITE(ILPRNT,2000)
      WRITE(ILPRNT,3000)
      WRITE(ILPRNT,1100) IDATE2,ICLOCK,VERNO ,MONTH ,JDATE ,TITLE
      WRITE(ICONSL,1001)
      WRITE(ICONSL,2000)
      WRITE(ICONSL,3000)
C
C *** THE FOLLOWING STATEMENT MAY CAUSE A WARINING MESSAGE WHEN EXECUTED
C *** ON IBM CMS WITH 'RUNHOST' PROCEDURE FILE.
C
      WRITE(ICONSL,1100) IDATE2,ICLOCK,VERNO ,MONTH ,JDATE ,TITLE
      WRITE(ICONSL,1002)
C
C ***********************************************************************
C
 1000 FORMAT(1H1,20(/))
 1001 FORMAT(20(/))
 1002 FORMAT(///)
 2000 FORMAT(
     *2X,'MA       RC  CC      CC   CCCCCCCC    CCCCCCCCC  CCCCCCCCCC'/,
     *2X,'MAZ     ZRC  CC      CC  CCCCCCCCCC  CCCCCCCCCC  CCCCCCCCCC'/,
     *2X,'MAZZ   ZZRC  CC      CC  CC      CC  CC              CC    '/,
     *2X,'MA ZZ ZZ RC  CC      CC  CC      CC  CC              CC    '/,
     *2X,'MA  ZZZ  RC  CCCCCCCCCC  CC      CC  CCCCCCCCC       CC    ')
 3000 FORMAT(
     *2X,'MA   Z   RC  CCCCCCCCCC  CC      CC   CCCCCCCCC      CC    '/,
     *2X,'MA       RC  CC      CC  CC      CC          CC      CC    '/,
     *2X,'MA       RC  CC      CC  CC      CC          CC      CC    '/,
     *2X,'MA       RC  CC      CC  CCCCCCCCCC  CCCCCCCCCC      CC    '/,
     *2X,'MA       RC  CC      CC   CCCCCCCC   CCCCCCCCC       CC    ')
 1100 FORMAT(///,2X,6HDATE: ,A8,3X,A8,//2X,
```

```
      1'VERSION ',F3.1,'  - DATED ',A4,',',A4,', 1985    ',
      *//,2X,20A4)
C
C ***********************************************************************
C
      RETURN
      END
```

Calls to the CPU clock for the execution time are also system dependent. SUBROUTINE TIME in the original PRIME version interfaces with the system's clock and returns the time in seconds as a double precision real variable.

```
C SUBROUTINE TIMER
      SUBROUTINE TIMER(CU)
C
C* * * * * *
C
C CPTIME RUNTIME FOR PRIME
C
C* * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 I(15),NUM
      NUM=15
      CALL TIMDAT(I,NUM)
      CU=I(7)+(I(8)/330.)
      RETURN
      END
```

On the CRAY, this operation is simplified and performed at a routine one level higher as:

```
C=SUBROUTINE=TIMOUT  UTILITY SUBROUTINE
      SUBROUTINE TIMOUT(IA1,IA2,IA3,IA4,IA5,IA6)
C
C ***********************************************************************
C
C     OBTAINS AND PRINTS CPU TIME
C
C ***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION NAME(6)
C
```

```
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,PI    ,LINE  ,LINE2
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JPRES ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JPRESS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS
      COMMON / TIMLOC / CPU0
C
C *** GET 'DELTA' VALUES - CALL TO CARY LIBRARY SUBROUTINE
C
      CALL SECOND (  CPU  )
C
      IF( CPU0 .EQ. 0.D0 ) CPU0 = CPU
                           CPU  = CPU - CPU0
C
      NAME(1) = IA1
      NAME(2) = IA2
      NAME(3) = IA3
      NAME(4) = IA4
      NAME(5) = IA5
      NAME(6) = IA6
C
      CALL LINES( 3 , 3 )
                        WRITE(ICONSL,1010) NAME,JINC,JITER,CPU
      IF(CPU.NE.0.D0) WRITE(ILPRNT,1000) NAME,JINC,JITER,CPU
C
 1000 FORMAT(//,2X,6A4,9HINCREMENT,I4,10H ITERATION,I3,8H CPTIME=,F9.3,
     1 4H SEC)
 1010 FORMAT(2X,6A4,9HINCREMENT,I4,10H ITERATION,I3,8H CPTIME=,F9.3,
     1 4H SEC)
      RETURN
      END
```

To be able to obtain a reasonably accurate estimate of the CPU time, it is recommended that these routines be modified for the particular system on which the MHOST program is installed.

1.0 CONTROL STRUCTURE

In this chapter, a description of major subprograms is given. This is preceded by a brief note on the architecture of the MHOST program. The third section here is devoted to the definition of control variables appearing in the common blocks.

1.1 Overview

The architecture of the MHOST code is schematically shown in Figure 1. The execution supervisor routine (SUBROUTINE HOST) controls several analysis modules in a consistent manner. In Version 4.2 of the MHOST code, a pair of subroutines, LETCMD and RUMCMD, have been added to check the consistency of the Parameter Data and to generate internal flags for the selection of analysis modules. The intention here is to stop users from combining features of the program package not designed to be used together.

```
C=SUBROUTINE=HOST   CALLED FROM 'HOST: MAIN' PROGRAM
      SUBROUTINE HOST
     1    (RWORK ,IWORK ,ISIZE ,VERSNO,MONTH ,JDATE )
C
C **********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C **********************************************************************
C
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / ADDVAL / ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
      COMMON / AUTOIN / CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
```

```
     1                 JADAP ,NCREEP,SCALE
      COMMON / PERIOD / JPEROD(   2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / POWER  / IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                 IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                 IPSF  ,IPSD
      COMMON / PULSES / IPULSE,KPULSE,IPDTIM,IPDFOR
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                 IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / ERRORS / IERR
      COMMON / MODSUP / IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      COMMON / HARMON / OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
     1                 ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                 ICCMAT
      COMMON / LOUBIN / JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
     1                 JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                 IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                 IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                 IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                 IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                 IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                 IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                 IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                 IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                 IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                 IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                 NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                 NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                 NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                 NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                 NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                 NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                 NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / BSECT  / IBSECT,JBSECT
      COMMON / INCCON / FACTOR,INCFLG(21)
      COMMON / MACHIN / IDP
      COMMON / SUBSTR / NLVSUB(   10),NFRSUB(   10)
      COMMON / SUBELM / ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
      COMMON / SHIFT  / ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
      DATA     IZ     /  1H         /
      DATA     MAXSUB /          10 /
C
C ***********************************************************************
```

```
C
      WRITE(ICONSL,2000) IZ
 2000 FORMAT(A1)
      CALL TIMOUT('  BE','GIN ','EXEC','UTIO','N,  ','    ')
C
C *** JOB PARAMETER INPUT ***********************************************
C
      CALL DATIN1
     1     (RWORK ,IWORK ,ISIZE ,VERSNO,MONTH ,JDATE ,NELEM ,NNODE ,
     2      NBC   ,NTIE  ,NMAX  ,NTRAN ,NTRAC ,NPOST ,NLVSUB,NFRSUB,
     3      NEXT  ,JDYN  ,JTEMP ,NPRINT,JREST ,JINC  ,NINC  ,JLOUB ,
     4      JINTER,JEXTRA,JWEIGH,NSTRBC,NTYPE ,MAXSUB,ILAST ,JSUB  ,
     5      NSUB  ,ISTAT ,IDYNM ,ITEST ,JOPTIM,JCREEP,JDIST ,NONISO,
     6      NDYNMD,IDYNMD,NPOSMD,ITHERM,JCONST,NDUP  ,JREPOT,JTANGE,
     7      JTHERM,DALPHA,DBETA ,DGAMMA,JEIGEN,JFORCE,JUTEMP,JUCOEF,
     8      JDISTS,JUHOOK,JDERIV,JUBOUN,JPEROD,NSBNC ,NCREEP,ATOLER,
     9      BTOLER,CTOLER,JPOST ,INTSTR,JBAND ,JFRONT,JDEFOR,NGMRS ,
     +      JEMBED,NBSECT,JDISP ,NSHIFT,NSUPER,JSUBRE,IFBFGS,NSPRI ,
     1      NDASH ,NMASS ,NSBFGS,IFSCNT,IFLINE,IFPRNT,NHARM ,OMEGH ,
     2      NBASE ,OMEGB ,ICOMPS,NPDPTS,NPULSE,IPCONJ,NSSPTS,NSHOCK,
     3      NPSPTS,NFDPTS,LDYN  ,JFDSXX,JISTIF,JCENTM,NHARD ,JFINIT,
     4      JLARGE,JFOLOW,JWKSLP,JISTRN,JCITER,JHRGLS,NDIMEN,JGRAM ,
     5      JPRES ,NMONIT)
C
C *** SET THE RUNSTREAM FLAGS FOR MULTIPLE OPTION EXECUTIONS ***************
C
      CALL LETCMD
     1     (NCOMPD,IFSTAT,IFSOLV,IFINTG,IFMODL,IFBCKL,IFSUPR,IFREQN,
     2      IFDUM1,IFDUM2,IFDUM3,IFDUM4,IERR  )
C
C *** BULK FINITE ELEMENT DATA INPUT ************************************
C
C     IF ( JREST ) IS SET ZERO, START FROM SCRATCH AND THE DATA IS READ
C                               FROM THE MAIN CARD READER CHANNEL
C     IF ( JREST ) IS NON-ZERO, THE RESTART FILE IS RECOVERED FROM THE
C                               RESTART FILE
C ***********************************************************************
C     FLAG 'IPFLG' IS USED FOR CONTROLING THE PRINTOUT FOR THE INITIAL
C     VALUES FOR TRANSIENT DYNAMICS
C
C     IPFLG = 0 ..... ONLY INITIAL CONDITION IS PRINTED
C             1 ..... ONLY THE REST OF THE MODEL DEFINITION DATA ARE
C                     PRINTED
C             2 ..... ALL OF THE INFORMATION IS PRINTED
C
```

```
                                        IPFLG = 1
      IF( IDYNM .EQ. 1 .AND. JDYN .EQ. 2 ) IPFLG = 2
C
C ***********************************************************************
C
      IF( JREST .EQ. 0 ) CALL BULKIN ( RWORK , IWORK , ISIZE , IPFLG )
C
C *** RESTART FILE INPUT *** NOT SUPPORTED IN VERSION 2.0 **************
C
      IF( JREST .NE. 0 ) CALL RESTRT ( RWORK , IWORK , ISIZE )
C
C ***********************************************************************
C
C     CALL FINITE ELEMENT SOLUTION DRIVER ROUTINES
C
C ***********************************************************************
C
      DO   5000   ICOMPD   =   1   ,NCOMPD
C
      CALL RUNCMD
     1    (ICOMPD,IFSTAT,IFSOLV,IFINTG,IFMODL,IFBCKL,IFSUPR,IFREQN,
     2     IFDUM1,IFDUM2,IFDUM3,IFDUM4,IERR  )
C
C ***********************************************************************
C
      IF( IFSTAT .EQ. 1 ) THEN
                              CALL STATIC( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFSOLV .EQ. 1 ) THEN
                              CALL FRONTS( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFMODL .EQ. 1 ) THEN
                              CALL MODAL ( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFSUPR .EQ. 1 ) THEN
                              CALL SUPER ( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFBCKL .EQ. 1 ) THEN
                              CALL BUCKLE( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFREQN .EQ. 1 ) THEN
                              CALL FREDOM( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFINTG .EQ. 1 ) THEN
                              CALL DYNAMT( RWORK , IWORK , ISIZE )
```

```
C
      ELSE IF( IFDUM1 .EQ. 1 ) THEN
                               CALL DYNAMC( RWORK , IWORK , ISIZE )
C
      ELSE IF( IFDUM2 .EQ. 1 ) THEN
                               CALL FESOLN( RWORK , IWORK , ISIZE )
C
      ELSE
            CALL QUIT
     1    ('NO D','RIVE','R SP','ECIF','IED ','    ', 1  )
C
      ENDIF
C
 5000               C O N T I N U E
C
      CALL QUIT
     1    ('END ','OF T','HE A','NALY','SIS ','    ', 0  )
C
C ************************************************************************
C
      RETURN
      END
```

The analysis modules represent the control structure for the incremental iterative algorithms implemented in the MHOST code. The source code with comments is designed to serve as the schematic flow chart of the computational process.

The schematic flow of the element and nodal data manipulations is coded in the element assembly submodules identified by the string ASSEM in the subroutine names which are entered from analysis modules. The element assembly submodules perform operations independent of element types and constitutive models. The assumptions introduced in the mechanics aspects of the formulations are explicitly coded at this level. These submodules call the librarian routines for elements and constitutive models.

The element librarian subprogram (SUBROUTINE DERIV) generates quantities unique to the element used in an analysis. The MHOST code uses standard finite element matrix notation as in ZIENKIEWICZ (1977). The element specific information returned from the librarian subprogram is the strain-displacement array referred to as the B matrix in previous writeups for 1987.

The constitutive equation librarian subprogram (SUBROUTINE STRESS and BMSTRS) is designed to incorporate the nodal storage of stress and strain values. The STRESS subroutine sets up the loop over the points at which

constitutive equations are evaluated. The current implementation is a nested double loop with the outer loop being over the nodes and the inner loop over the integration layers through the thickness. Note that the conventional displacement method can be recovered by restructuring this loop in conjunction with a few minor modifications in the core allocation for stresses and strains.

The librarian subprogram utilizes the constitutive equation package (SUBROUTINE NODSTR) from which individual subprograms for initial strains, stress recovery and the material tangent are called. The librarian subprogram also controls the pre-integration of stresses, strains and the material tangent over thickness for the shell element.

Note that application of the constitutive equation represents one of the most costly operations in nonlinear finite element computations. An attempt is made in MHOST to optimize the execution of this process which occurs, for example, once every iteration during the recovery of the residual vector. At the beginning of each increment, this process may be executed to evaluate the material tangent and to determine the contribution of initial strain terms which are often necessary for proper displacement pre-conditioning.

Except for a small amount of information related to convergence of the iterative solution, all report generation is performed at the end of an increment. Optionally, post-processing and restart files are written at the end of user-specified increments. Generic reporting subprograms are called from analysis modules inside the increment loop.

1.2 Execution Supervisor and Analysis Drivers

This section describes major subprograms in the MHOST package. For the sake of clarity no listings are attached in this section. Also this section does not discuss the arguments and common blocks appearing in each subroutine.

Execution Supervisor

MAIN PROGRAM - declares work space in blank common as an integer array. Also defines system parameters which are machine independent. Then passes control to actual execution supervisor SUBROUTINE HOST.

SUBROUTINE HOST - controls the sequence of execution for the analysis modules. First, this routine executes the control parameter data reader SUBROUTINE DATIN1 and checks for consistency by entering SUBROUTINE LETCMD and RUNCMD. The current structure of the code allows certain combinations

of two analysis modules to be executed sequentially. Analysis modules called from the execution supervisor are:

SUBROUTINE STATIC for a quasi-static incremental iterative solution.

SUBROUTINE DYNAMT for a transient time integration of the dynamic equilibrium equations in an incremental-iterative manner.

SUBROUTINE MODAL for an eigenvalue extraction in vibration mode analysis. This subsystem may be executed after quasi-static analysis for modal analysis of prestressed structures.

SUBROUTINE BUCKLE for an eigenvalue extraction in buckling load calculations. This subsystem is executable only after quasi-static analysis.

SUBROUTINE FRONTS for a quasi-static incremental iterative solution by the out-of-core frontal approach. Note that certain options are not available in this subsystem.

SUBROUTINE SUPER for linear dynamic response calculations by the method of mode superposition.

Input Data Reader

There are three major subprograms:

SUBROUTINE DATIN1 - Reads and interprets the parameter data input called by the execution supervisor. All the default values for control variables are set in this routine.

SUBROUTINE BULKIN - Reads and interprets the finite element model definition data and prints the mesh and loading data for the initial increment (number 0). The bulk data reader is entered before the execution of analysis modules. This subroutine utilizes the following lower level routines:

SUBROUTINE INITI1 for memory allocation of integer workspace in blank common to store nodal and element data.

SUBROUTINE DATIN2 for model data input.

SUBROUTINE DATOU1 for reporting model definition data.

SUBROUTINE CHKELM for the detection of clockwise element connectivity definition which results in a negative Jacobian.

This routine automatically corrects the connectivity table to a counterclockwise direction.

SUBROUTINE SUBDIV for memory allocation of subelement mesh data and automatic mesh generation of subelements.

SUBROUTINE INCRIN - Reads and interprets the loading and constraint data for each increment. This routine is invoked by individual analysis modules from inside the loop over the increments.

SUBROUTINE DATIN3, a small subset of the bulk data reader SUBROUTINE DATIN1 is used for actual operations including initialization of arrays at the beginning of an increment.

Algebraic Operation Subsystem

There are three groups of routines for the profile solver, frontal solver and eigenvalue extraction, respectively. The profile solution package consists of:

SUBROUTINE COMPRO - Sets up the integer array for the profile of global stiffness equations to be stored in a profile form.

SUBROUTINE ASSEM5 - Assembles the element stiffness equations into the global equation system stored in a profile form.

SUBROUTINE SOLUT1 - Controls the iterative solution processes including the vector update required for the quasi- and secant-Newton iterations.

SUBROUTINE DECOMP - Factorizes the global stiffness equations stored in a profile form.

SUBROUTINE SOLVER - Performs the back substitution and generates the update vector for the incremental displacement.

The frontal solution package consists of:

SUBROUTINE FRONTW - Estimates the front matrix size to be accommodated in the core memory.

SUBROUTINE INITFR - Allocates memory for the work space required for the frontal solution.

SUBROUTINE PRFRNT - Sets up the elimination table for the frontal solution.

SUBROUTINE FRONTF - Assembles and factorizes the global stiffness equation simultaneously.

SUBROUTINE FRONTB - Performs the back substitution and generates the updates for the displacement vector.

SUBROUTINE FRONTR - Controls the iterative solution processes including the vector update required for the quasi- and secant-Newton iterations. Calls SUBROUTINE FRONTB for the incremental displacement update vector.

SUBROUTINE VDSKIO - Controls the data stream stored in the in-core buffer area and the actual out-of-core storage devices.

The eigenvalue analysis package consists of:

SUBROUTINE EIGENV - Controls the execution of the eigenvalue extraction subsystem.

SUBROUTINE INIMOP - Initializes the array for eigenvectors.

SUBROUTINE SUBSPC - Performs the subspace iteration and generates a specified number of eigenvalues and eigenvectors.

SUBROUTINE JACOBI - Solves the eigenvalue problem in subspace by the Jacobi iteration.

There are a number of subprograms used commonly by the braic Operation Subsystem:

SUBROUTINE STRUCT - Controls the memory allocation for the global algebraic manipulations at the beginning of every increment.

SUBROUTINE INITI2 - Allocates memory required for the storage of global stiffness matrix and other vectors required in the linear algebraic manipulation of finite element equations.

SUBROUTINE LINESR - Calculates the search distance when the line search option is turned on.

Element Assembly Submodules

These are subprograms that construct vectors and matrices appearing in the algorithmic description of mixed iterative process discussed in the previous subsections.

SUBROUTINE ASSEM1 - Assembles the displacement stiffness matrix for preconditioning purposes. All the kinematic and constitutive options are tested in this module.

SUBROUTINE ASSEM2 - Assembles the coefficient matrix for transient time integration by the Newmark family of algorithms. This routine has evolved from SUBROUTINE ASSEM1 and contains all the same options.

SUBROUTINE ASSEM3 - Assembles the coefficient matrix for quasi-static analysis using the frontal solution subsystem. Large displacement, stress stiffening and centrifugal mass terms are not available in this package.

SUBROUTINE ASSEM4 - Calculates the nodal strain and recovers the residual vector in a mixed form. The subelement solution package is entered from this subprogram.

Element Loop Structure and Library Routines

In the element assembly submodules, element arrays are generated in the loops over elements. The protocol for accessing the element library is designed and implemented to involve a sequence of subroutine calls:

SUBROUTINE ELVULV - Sets up the current element parameters (See Table 2 for variables and values) from the element library table.

SUBROUTINE CNODEL - Pulls out quantities for the current element from the global nodal array and restores them in the element workspace. Coordinate transformations necessary for beam and shell elements are performed in this subprogram.

SUBROUTINE DERIV - Sets up the displacement-strain matrix for the current element by calling the element library subroutines. Those are:

SUBROUTINE BPSTRS for plane stress elements, types 3 and 101.

SUBROUTINE BPSTRN for plane strain elements, types 11 and 102.

SUBROUTINE BSOLID for three-dimensional solid element, type 7.

SUBROUTINE BSHELL for three-dimensional shell element, type 75.

SUBROUTINE BAXSYM for axisymmetric solid-of-revolution elements, types 10 and 103.

SUBROUTINE BTBEAM for linear Timoshenko beam element, type 98.

SUBROUTINE BASPST for the assumed stress plane stress element, type 151.

SUBROUTINE BASPSN for the assumed stress plane strain element, type 152.

SUBROUTINE BASSOL for the assumed stress three-dimensional solid element, type 154.

SUBROUTINE UDERIV - A slot for user coded element B matrix routine.

The following subprograms are used to calculate terms appearing in the finite element equations:

SUBROUTINE LMPMAS - Calculates nodal weight factor for the strain projection.

SUBROUTINE STIFF - Performs matrix triple products to assemble the element stiffness matrix and the element load vector associated with the initial strain terms.

SUBROUTINE STRAIN - Calculates element strain at specified sampling points and projects to nodes.

SUBROUTINE CNSMAS - Assembles the consistent mass matrix for modal and transient analysis.

SUBROUTINE INITST - Generates initial stress terms for quasi-static, buckling and modal analysis of prestressed structures.

SUBROUTINE CENMAS - Evaluates the centrifugal mass terms for rotating structures at speed.

SUBROUTINE RESID - Calculates the element residual vector for the global element.

SUBROUTINE SUBFEM - Performs the subelement solution and calculates the element residual vector for the subelement mesh.

SUBROUTINE RELDFG - Calculates the relative deformation gradient at the element sampling points and projects to nodes.

SUBROUTINE RESDYN - Calculates the contribution of mass and damping

terms in the element residual vector when the transient dynamics option is used.

Material Library

A system of subroutines is included in the MHOST code which covers a wide range of material models and initial strain assumptions:

SUBROUTINE SIMPLE - Integrates the stress over the increment assuming the elastic-plastic response of the material is modeled by the total secant modulus approach. Also generates the material modulus matrix.

SUBROUTINE PLASTS - Integrates the stress over the increment and calculates the plastic strain by using the radial return algorithm.

SUBROUTINE PLASTD - Calculates the consistent elastic-plastic modulus if the incremental equivalent plastic strain is positive.

SUBROUTINE WALKEQ - Integrates the Walker unified creep plasticity constitutive equation and also generates a material modulus based on the temperature dependent elasticity assumption.

SUBROUTINE LELAST - Calculates the stress for the constant material modulus given as data.

SUBROUTINE THRSTN - Calculates the thermal strain.

SUBROUTINE CRPSTN - Calculates the creep strain.

Report Generation Subsystem

A system of subroutines is included in the MHOST code to generate reports on the line printer image file and the formatted post-processing file readable by MENTAT, Version 5 and up (a proprietary interactive pre- and post-processor program developed by and available from MARC Analysis Research Corporation). Note that the format of the post-processing file is similar to that of the MARC General Purpose Finite Element Program, Version K.1 and the commercially available post-processing packages compatible with this program can easily be modified for use with MHOST.

SUBROUTINE PRININ - Reads and restores the user instructions for the print options. This routine is entered from the model data reader and/or the incremental reader. This subprogram interprets user instructions including the type of information, the locations (either nodes or element integration points) at which the information is

printed and specified by character strings, and the range of nodes or elements specified by integer numbers.

SUBROUTINE PRINOU - Writes the reports on the line printer image file. A loop is constructed over the instruction set restored by PRININ.

SUBROUTINE PRINSU - Generates the reports for the subelement solution on the line printer image file. The same control structure as the global solution writer PRINOU is implemented. This routine also follows the instruction set generated by PRININ.

SUBROUTINE POSTOU - Writes formatted records on the post-processing file, Fortran Unit 19. All the information generated by the analysis is written. The user has no control over the records to be written on this file. The nodal values of stress and strain invariants and components are written in this subroutine.

SUBROUTINE POSTEN - Packs the nodal value record buffer and writes out the record when the buffer is filled.

1.3 Control Variables

The control variables effective globally through a number of subprograms are stored in the following common blocks:

ADDVAL	Variables related to additional point masses, dashpots and springs.
ALGEM	Fortran unit numbers and record number counters. Also the value of p is stored in this common block.
AUTOIN	Real variables used for automatic load increments. Also control variables for adaptive load increments by the arc length method are stored here.
BODYFR	A real array to store the direction of the body force vector for gravity loading or the position of two points defining the axis of rotation for centrifugal loading.
BSECT	Pointers and counters for beam section definitions.
CONTRO	Control variables and binary switches to activate options built into the MHOST program.
COMPND	Flags for sequential execution of multiple analysis drivers.

COUNT	Counters for the line on the current page of the line printer image file.
CTITLE	The title line and the date and time when the execution is initiated.
DAMP	A damping factor array for transient dynamic analyses.
EIGEN	Counters for eigenvalue extraction by the subspace iteration method.
MODSUP	Counters for transient response calculation by vibration mode superposition.
ELEMEN	A element type indicator and flags for element array manipulations.
ELTYP	Counters for the element variables which are currently being processed. The entries in this common block are redefined every time the element loop is incremented.
ERRORS	Data input error counter.
FREE	Working arrays for the free format reader.
HARMON	Control variables for harmonic analysis
INCCON	Variables for the incremental data reader.
MACHIN	Machine dependent ratio of the integer and real word lengths.
MAXIMA	Maximum number of element variables appearing in the current problem.
LOUBIN	Flags for the selection of integration processes for various element arrays.
PAGCNT	Counters for the page numbers of the line printer image output file.
PARAM	Mesh parameters for the total number of data entries.
PERIOD	Parameters to define time periodic loading and boundary conditions.

POSTPN Control variables used for generation of the post-processing file.

POWER Control variables for the power spectrum.

PULSES Counters and pointers for pulse leads in dynamic analysis.

RESULT Control variables for generation of the line printer image output file.

SUBELM Control variables for the subelement analysis option.

SUBTYP Element variables for the subelement analysis option. The information stored in this common block is the subelement counterpart of ELTYP and entries are updated every time the loop over the subelements is incremented.

SUBSTR Space allocated for the control variables of the substructure option (not used in Version 4.2).

SHIFT Control variables for the power-shift option in the eigenvalue extraction for modal analysis.

TIME Time parameters for transient dynamics and rate dependent calculations.

TIMLOC The initial value for the CPU clock.

TMARCH Parameters to define the time integration operator.

TRANSF Local coordinate transformation arrays. Typically updated in an element-by-element manner.

TOLER Error and tolerance for the global iterative solution.

VRTDSK Control variables for the out-of-core solution by the frontal process.

ZPRINT Control variables for a definition of a record in the line printer output file.

Control variables which play a significant role are summarized below:

Variable Name	Common Block	Initial Value	Content

ICREAD	ALGEM	5	Fortran unit number for the main data input device.
ILPRNT	ALGEM	6	Fortran unit number for the line printer output file (Execution log).
JLPRNT	ALGEM	6	Fortran unit number for the line printer output file (Bulk report).
ICONSL	ALGEM	1	Fortran unit number for the command input and the execution log output.
IPOSTF	ALGEM	19	Fortran unit number for the formatted post-processing file.
ISCRAF	ALGEM	10	Fortran unit number for the working on-line storage (binary).
IPOSTB	ALGEM	9	Fortran unit number for plotting data output file (binary - not used in Version 4.2).
IRSTRT	ALGEM	8	Fortran unit number for binary restart file.
JCREAD	ALGEM	12	Fortran unit number for alternate input data file.
LINE	ALGEM	0	Line counter for the output unit ILPRNT.
LINE2	ALGEM	0	Line counter for the output unit JLPRNT.
CURPER	AUTOIN	0.0	Load factor of the current arc-length iteration step.
TOTPER	AUTOIN	0.0	Total load factor including the current step of arc-length iteration.
ARCLEN	AUTOIN	0.0	The arc-length.
JADP	AUTOIN	0	Flag for the adaptive load increment option.

Variable Name	Common Block	Initial Value	Content
NCREEP	AUTOIN	0	Maximum number of updates for the adaptive time step control for creep strain.
JEND	CONTRO	0	Flag for end-of-iteration.
JITER	CONTRO	0	Current iteration step of the global solution.
JTEMP	CONTRO	0	Flag for temperature loading.
JPRINT	CONTRO	0	Full line printer report generation interval for incremental analysis.
JP	CONTRO	0	Flag for pressure loading.
JSUB	CONTRO	0	Flag for substructure.
JINC	CONTRO	0	Current increment.
JREST	CONTRO	0	Flag for restart jobs.
JSAVE	CONTRO	0	Flag for check-point job to write the restart file.
JREDIM	CONTRO	0	Flag for memory allocation of global arrays for the next solution step.
JAUTO	CONTRO	0	Flag for automatic load incrementation.
JPOST	CONTRO	0	Flag for post file generation.
JBACK	CONTRO	0	Flag for back substitution to avoid the global stiffness assembly and factorization.
JOPTIM	CONTRO	0	Number of Cuthil-McGee iterations for bandwidth optimization.
JCREEP	CONTRO	0	Flag for the creep strain option.

Variable Name	Common Block	Initial Value	Content
JDIST	CONTRO	0	Flag for the distributed load option.
JCONST	CONTRO	0	Selector for the constitutive equation = 0 for linear elasticity 1 for secant elasticity/ simplified plasticity 2 plasticity 3 unified creep plasticity
JDYN	CONTRO	0	Flag for the dynamic analysis option.
NONISO	CONTRO	0	Flag for anisotropic material response.
ITHERM	CONTRO	0	Flag for temperature dependent material response.
NXSTAT	COMPND	0	Flag for the static analysis module as the driver for the second step of the analysis.
NXSOLV	COMPND	0	Flag for the optional frontal solution for the second step of the analysis.
NXINTG	COMPND	0	Flag for the direct time integration for the second step of the analysis.
NXMODL	COMPND	0	Flag for modal analysis for the second step.
NXBCKL	COMPND	0	Flag for buckling analysis for the second step.
NXSUPR	COMPND	0	Flag for modal super-position for the second step of the analysis.

Variable Name	Common Block	Initial Value	Content
TITLE	CTITLE	' '	An array to store the character string representing the title of the current analysis job.
IDYNMD	EIGEN	0	Number of eigenvalues and eigenvectors to be extracted.
IPTAR	EIGEN	0	Pointer to the workspace required for the subspace iteration. If the BFGS update is invoked in the static analysis, this variable is used as a pointer for an update vector.
IPTBR	EIGEN	0	Pointer to the workspace required for the subspace iteration. If the BFGS update is invoked in the static analysis, this variable becomes a pointer for one of the update vectors.
IC	ELEMEN	0	Current element type. This variable is updated when the element loop is incremented.
IEL	ELEMEN	0	Current element number.
IDF	ELEMEN	0	Number of d.o.f. per element being processed.
JLAW	ELEMEN	0	Constitutive equation type:
		= 2	plane stress
		= 3	plane strain
		= 4	axi-symmetric solid
		= 5	three dimensional continuum
		= 6	shell
		= 7	beam
IPATH	ELEMEN	0	Flag for the integration method options for assembly and recovery.

Variable Name	Common Block	Initial Value	Content
IASSEM	ELEMEN	0	Flag to indicate element assembly operations.
NELCRD	ELTTP	0	Number of coordinate entries per node for the current element.
NELNFR	ELTYP	0	Number of d.o.f. per node for the current element.
NELNOD	ELTYP	0	Number of nodes per current element.
NELSTR	ELTYP	0	Number of generalized stress/strain components per node for the current element.
NELPR	ELTYP	0	Number of distributed load types for the current element.
NELINT	ELTYP	0	Number of integration points for the current element.
NELLV	ELTYP	0	Number of d.o.f. per current element.
NELLAT	ELTYP	0	Number of integration layers for the shell element to calculate resultant quantities.
NDI	ELTYP	0	Number of direct strain/stress components for the current element.
NSHEAR	ELTYP	0	Number of shear strain/stress components for the current element.
NELCMP	ELTYP	0	Number of LDamina strain/stress components (differs from NELSTR only for shells).
IA	FREE	' '	Buffer for the card image read from ICREAD file.

Variable Name	Common Block	Initial Value	Content
MAXCRD	MAXIMA	1	Maximum number of coordinate entries in the current mesh..
MAXNFR	MAXIMA	2	Maximum number of d.o.f. per node in the current mesh.
MAXNOD	MAXIMA	1	Maximum number of nodes per element in the current mesh.
MAXCHR	MAXIMA	1	Maximum number of material data entries for the element definition in the current mesh.
NSUMAX	MAXIMA	0	Maximum number of substructures (not-used in version 4.2)
MAXBSP	MAXIMA	6	Maximum number of entries in the beam section definition in the mesh.
MAXDMT	MAXIMA	1	Maximum dimension of the constitutive resultant matrix.
MAXFRN	MAXIMA	0	Maximum front matrix size (valid when the frontal solution option is invoked).
MAXEAN	MAXIMA	0	Maximum number of entries in the inverse connectivity table. The value is set in INITI2.
MAXBET	MAXIMA	0	Maximum number of entries in the element strain-displacement matrix for all the integration points.
JLOUB	LOUBIN	1	Not used in Version 4.2.
JINTER	LOUBIN	3	Selector for the integration rule used to generate the strain-displacement equation.
		= 1	Reduced integration
		= 2	Selective integration

			(one point shear sampling)
		= 3	Selective integration with the element coordinate transformation.
JEXTRA	LOUBIN	1	Selector for the integration rule used in the residual force calculation.
		= 1	Full Gaussian quadrature
		= 2	One point reduced integration
JWEIGH	LOUBIN	3	Selector for the integration rule used to generate the stiffness equations.
		= 1	Full Gaussian quadrature
		= 2	Selective integration (one point shear sampling without coordinate transformation).
		= 3	Selective integration with the element coordinate transformation.
NPAGE1	PAGCNT	0	Page counter for the line printer output on ILPRNT.
NPAGE2	PAGCNT	0	Page counter for the line printer output on JLPRNT

Variable Name	Common Block	Initial Value	Content
NTYPE	PARAM	0	Number of different element types in the current mesh.
NELEM	PARAM	0	Total number of elements in the current mesh.
NBC	PARAM	0	Total number of displacement boundary conditions in the current mesh. The user specified value in the parameter data section is used for memory allocation. The value is updated to the actual value when the boundary displacement data is

			read in the model data.
NNODE	PARAM	0	Total number of nodes in the present mesh.
NTIE	PARAM	0	Total number of tying equations in the present mesh.
NMAX	PARAM	0	Maximum number of terms in the tying equations in the present mesh.
NTRAN	PARAM	0	Total number of nodal coordinate transformations specified by the user in the current mesh.
NSBFGS	PARAM	0	Number of update vectors for the BFGS quasi-Newton method.

2.0 DATA STORAGE SCHEME

This chapter discusses the dynamic core allocation scheme implemented in the MHOST code. The first section outlines the design principle and is followed by the code actually allocating memory in the second section. The third section is devoted to the definition of pointers.

2.1 Overview

A dynamic core allocation strategy is implemented in the MHOST code. An integer array is defined as an entry to blank common. This array is used as the working storage. Except for the frontal solution and transient dynamic analysis options, all the information required for the finite element computation is stored in this work space. First parameters defining the number of variables for element types in use is copied from the array residing in SUBROUTINE TYPEIN.

The first section of this space stores the nodal variables and element arrays. The amount of memory required to store these data is found from the number of nodes, elements and attributes given by the user as part of the parameter data. This portion of working storage and information stored therein is often referred to as the finite element database. The pointers for the finite element database are defined in SUBROUTINE INITI1 before reading the model data section of the input data file. The model data is read directly into the space allocated for the finite element database.

In finite element computations, space is required to store arrays where size varies depending on the analysis type and mesh topology. For instance, in most of the analysis options, the global stiffness matrix is stored in a profile form. After the model data is read, the mesh topology (element connectivity) is swept through and column heights are calculated for each degree of freedom in SUBROUTINE COMPRO. The pointers for the global arrays are then calculated in INITI2.

For the local-global analysis inherent to the subelement procedure, an indirect accessing scheme is implemented. When the subelement option is invoked by the parameter data, element arrays to store subelement data pointers are prepared as a part of the finite element database. The pointers are calculated only for the global elements which are divided into subelements. The mechanism of this indirect addressing scheme is implemented in the memory allocation routine INITSE and the subelement solution driver routine SUBFEM.

2.2 Memory Allocation Subprograms

The following SUBROUTINE TYPEIN substitutes the element characteristics array into the workspace:

```
C=SUBROUTINE=TYPEIN  CALLED FROM SUBROUTINE 'DATIN1'
      SUBROUTINE TYPEIN
     1     (IWORK ,RWORK ,IERR  ,NTYPE ,ILAST ,ILPRNT,NDIMEN)
C
C *********************************************************************
C
C     ROUTINE FOR OBTAINING INPUT FOR DIFFERENT TYPES OF ELEMENTS
C
C     IERR    ERROR FLAG
C     NTYPE   NUMBER OF ELEMENT TYPES
C     ILAST   LAST WORD FOR INPUT
C     ILPRNT  OUTPUT DEVICE
C     NDIMEN  DIMENSIONS OF THE ANALYSIS PROBLEM
C
C *********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C *********************************************************************
C
      DIMENSION IWORK ( 1 )  ,  RWORK ( 1 )
      DIMENSION ISTORE( 13 , 16 )
```

```
C
C ********************************************************************
C     PARAMETERS TO CHARACTERIZE ELEMENTS AVAILABLE IN THE HOST PROGRAM
C ********************************************************************
C
C    NO.TYPE.NCRD.NDOF.NODE.NSTR.NCHR.NINT.NLLV.NLAY.NCMP. NDI.NSHR.JLAW.
      DATA ISTORE
     1/   9,   3,   3,   2,   1,   6,   2,   0,   1,   1,   1,   0,   1,
     2    3,   2,   2,   4,   3,   6,   4,   3,   1,   3,   2,   1,   2,
     3   11,   2,   2,   4,   4,   6,   4,   3,   1,   4,   3,   1,   3,
     4   10,   2,   2,   4,   4,   6,   4,   3,   1,   4,   3,   1,   4,
     5    7,   3,   3,   8,   6,   6,   8,   3,   1,   6,   3,   3,   5,
     6   75,   7,   6,   4,   9,   6,   4,   4,   5,   5,   2,   3,   6,
     7   72,   7,   6,   4,   9,   6,   4,   4,   5,   5,   2,   3,   6,
     8   98,   6,   6,   2,   6,   6,   1,   3,   1,   6,   3,   3,   7,
     9  101,   2,   2,   9,   3,   6,   9,   3,   1,   3,   2,   1,   2,
     +  102,   2,   2,   9,   4,   6,   9,   3,   1,   4,   3,   1,   3,
     1  103,   2,   2,   9,   4,   6,   9,   3,   1,   4,   3,   1,   4,
     2  104,   3,   3,  27,   6,   6,  27,   3,   1,   6,   3,   3,   5,
     3  151,   2,   2,   4,   3,   6,   4,   3,   1,   3,   2,   1,   2,
     4  152,   2,   2,   4,   4,   6,   4,   3,   1,   4,   3,   1,   3,
     5  153,   2,   2,   4,   4,   6,   4,   3,   1,   4,   3,   1,   4,
     6  154,   3,   3,   8,   6,   6,   8,   3,   1,   6,   3,   3,   5/
C    NO.TYPE.NCRD.NDOF.NODE.NSTR.NCHR.NINT.NLLV.NLAY.NCMP.NNDI.NSHR.JLAW.
C
C ********************************************************************
C
                 NTYPES       =       16
                 JKEY         =       -1
C
C ********************************************************************
C     FREE FORMAT READER FOR ELEMENT TYPE DEFINITION
C ********************************************************************
C
  101 CONTINUE
      CALL FREFOR(IWORK(ILAST+1),RWORK(ILAST+1),13,0,0,IERR,JKEY)
C
                           IF( JKEY  .EQ. 1 ) GO TO  501
                           IF( NTYPE .EQ. 0 ) GO TO  301
C
      DO     201     J     =     1     ,      NTYPE
C
      K = ILAST + 1 - 13*J
      IF(IWORK(ILAST + 1).NE.IWORK(K)) GO TO  201
      CALL COPYIN(IWORK(ILAST + 1),IWORK(K), 13 )
```

```
      LL = IWORK(K)
                                                         GO TO  401
  201 CONTINUE
  301 CONTINUE
C
      DO     351    LL     =      1     ,      NTYPES
C
      IF( ISTORE(1,LL).EQ.IWORK(ILAST + 1 ))            T H E N
           CALL COPYIN(ISTORE(1,LL),IWORK(ILAST + 1), 13 )
      IF( ISTORE(13,LL).LE.4 ) NDIMEN = 2
                                                                E N D I F
   22 CONTINUE
C
  351                C  O  N  T  I  N  U  E
C
      NTYPE    =    NTYPE     +      1
      ILAST    =    ILAST     +     13
C
  401 CONTINUE
                                                         GO TO  101
  501 CONTINUE
C
C ***********************************************************************
C
      RETURN
      END
```

As seen in the following example, the element data copied by the above subprogram is accessed every time the element loop is incremented:

```
C
C ***********************************************************************
C     FIRST ELEMENT LOOP: FORM THE 'LUMPED' MASS MATRIX
C                         ALSO FORM NODAL NORMALS FOR SHELL ELEMENTS
C ***********************************************************************
C
                                   CALL NUL(REA(IMASNO),NNODE)
                                   CALL NUL(REA(IGSINO),NNODE*NELSTR)
                                   CALL NUL(REA(IGEPNO),NNODE*NELSTR)
C
C ***********************************************************************
C
      CALL NUL( REA(IEQPSI) ,         NELLAY*NELNOD )
      CALL NUL( REA(IIPSNO) , NELCMP*NELLAY*NELNOD )
C
```

```
C >>> START THE FIRST ELEMENT LOOP >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      DO 5500 IELCNT = 1 ,NELEM
              IC     = INT( ITYP + IELCNT - 1 )
              I1     = INEL + MAXNOD*( IELCNT - 1 )
C ***
         IF( IC  .EQ.  0 )                              GO TO 5500
C ***
      CALL ELVULV( INT , IC , IERR )
      CALL CNODEL( REA ,INT ,INT(I1) ,CTRANS ,IELCNT ,IC ,-1 ,-1)
C
      CALL LMPMAS
     1   (REA(IMASNO),REA(IDET)  ,INT(I1)   ,NELNOD    ,NELINT      ,
     2    NNODE       ,IC        ,REA(ICH)  ,REA(ICOR) ,NELCRD      ,
     3    IELCNT      ,NELEM     ,REA(INOD) ,MAXCRD    ,CTRANS      ,
     4    JGRAM       )
C
      IF (JPRES .GT. 0) CALL BOUNDN
     1   (REA(IBNORM),REA(ICOR  ),INT(I1    ),IELCNT     ,IC         ,
     2    NNODE       ,NELNOD     ,NELCRD     )
C
 5500 CONTINUE
C
C <<< CLOSE THE FIRST ELEMENT LOOP <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C
C ... NORMALIZE THE NODAL NORMALS FOR BEAMS AND SHELLS
C
      IF (JLAW.EQ.6 .OR. JLAW.EQ.9) CALL NRMNRM(REA(INOD),MAXCRD,NNODE)
C
C ***********************************************************************
C
 8500                        CONTINUE
```

SUBROUTINE ELVULV as shown below accesses the entries defined by SUBROUTINE TYPEIN:

```
C=SUBROUTINE=ELVULV  CALLED BY SUBROUTINES 'ASSEM1','ASSEM4'
      SUBROUTINE ELVULV
     1    (IWORK ,IC1   ,IERR  )
C
C ***********************************************************************
C
C     EXTRACTS INFORMATION ABOUT A CERTAIN ELEMENT TYPE
C
C     IC1     ELEMENT TYPE
```

```
C     IERR    ERROR FLAG
C
C ********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
              REAL*4   RWORK
C
C ********************************************************************
C
      DIMENSION IWORK(1)
C
C ********************************************************************
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / ELTYP  / NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELPR ,
     1                  NELINT,NELLV ,NELLAY,NDI   ,NSHEAR,NELCMP
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                  IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
C
C ********************************************************************
C
```

```
      DO 5000 II = 1 ,NTYPE
C
      I1 = IWORK(IELPRM + 13*(II - 1))
      IF(I1 .EQ. IC1) GO TO 8000
 5000 CONTINUE
      CALL LINES( 1 , 1 )
      WRITE(ILPRNT,9999) IC1
      WRITE(ICONSL,9999) IC1
 9999 FORMAT(2X,28H***ERROR***     ELEMENT TYPE,I5,12H NOT DEFINED)
C
C **********************************************************************
C
      IERR = IERR+1
C
C **********************************************************************
C
 8000 CONTINUE
      IS1    = IELPRM + 13*(II - 1) + 1
      NELCRD = IWORK(IS1)
      NELNFR = IWORK(IS1+1)
      NELNOD = IWORK(IS1+2)
      NELSTR = IWORK(IS1+3)
      NELCHR = IWORK(IS1+4)
      NELINT = IWORK(IS1+5)
      NELPR  = IWORK(IS1+6)
      NELLAY = IWORK(IS1+7)
      NELCMP = IWORK(IS1+8)
      NDI    = IWORK(IS1+9)
      NSHEAR = IWORK(IS1+10)
      JLAW   = IWORK(IS1+11)
      NELLV  = NELNFR*NELNOD
      IDF    = NELNFR*NELNOD
C
C **********************************************************************
C
C     SPECIAL TREATMENT FOR THE NINE POINT INTEGRATION OF THE QUADRATIC
C     LAGRANGIAN ELEMENTS ( ONLY IN CASE OF MATRIX ASSEMBLY )
C
      IF ( IASSEM .EQ. 0 .AND. NELINT .EQ. 9 ) NELINT  =  4
C
C **********************************************************************
C
      IF (JPRINT.LT.3 .OR. NBAND.EQ.0) GO TO 9000
C
      CALL LINES( 4, 4 )
```

```
      WRITE(ILPRNT,1920) IEL,IC1    ,
     1          NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELINT,
     2          NELPR ,NELLAY,NELCMP,NDI    ,NSHEAR,JLAW
 1920 FORMAT(/,1X,14HELEMENT NUMBER,I4,6X,6H  TYPE,I6,6X,10HPARAMETERS,
     1        /,7X,6HNELCRD,I6,6X,6HNELNFR,I6,6X,6HNELNOD,I6,
     2          6X,6HNELSTR,I6,6X,6HNELCHR,I6,6X,6HNELINT,I6,/,
     3          7X,6HNELPR ,I6,6X,6HNELLAY,I6,6X,6HNELCMP,I6,
     4          6X,6HNDI    ,I6,6X,6HNSHEAR,I6,6X,6HJLAW   ,I6)
C
 9000 R E T U R N
      END
```

In the element loops, information related to the element being processed is pulled out from the global finite element database and substituted into the element work arrays. The following SUBROUTINE CNODEL is called before the element data manipulation:

```
C=SUBROUTINE CNODEL= CALLED BY 'ASSEM1','ASSEM4','MASMAT'
C                              'DIV2Q2','ASSEM3','INITST','NEWRHS',
C                              'PERTRB','RESTOR'
      SUBROUTINE CNODEL
     1     (REA    ,INT    ,ISLV   ,CTRANS,IEL    ,IC     ,IFLAG ,ISTEP )
C
C **********************************************************************
C
C     EXTRACTS ELEMENT QUANTITIES AND UPDATES GEOMETRY
C
C     ISLV    CONSTITUTIVE LAW VALUES
C     IEL     POINTER TO ELEMENT NUMBER
C
C **********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
               REAL*4 REA
C
      DIMENSION REA(1),INT(1),ISLV(1),CTRANS(3,3)
      DIMENSION VTRANS(2,2),TTRANS(3,3),ETRANS(3,3)
C
C **********************************************************************
C
      COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI     ,LINE   ,LINE2
      COMMON / CONTRO / JEND   ,JITER ,JTEMP ,JPRINT,JP     ,JSUB   ,
```

```
     1                   JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                   JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                   NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                   JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                   JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                   JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                   IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                   IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                   JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                   JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / ELTYP  / NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELPR ,
     1                   NELINT,NELLV ,NELLAY,NDI   ,NSHEAR,NELCMP
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                   MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                   MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                   MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                   MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                   NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                   NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                   NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                   NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                   NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                   NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                   NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                   ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                   IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                   IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                   IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                   IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                   IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                   IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                   IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                   IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                   IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                   IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                   IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                   IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                   IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                   KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
```

```
     4                    KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                    IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                    KDMAT ,KDMTNO,KTDSNO,KITDST,IXM    ,IXC    ,
     7                    IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                    IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON   ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                    IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                    KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                    KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      COMMON / MACHIN / IDP
C
C ************************************************************************
C
C
      NELST2  = NELSTR * NELSTR
      MAXST2  = MAXSTR * MAXSTR
      IPP     = IPRES + (IEL-1) * MAXPRS * IDP
      MAXDIM  = MAXCMP * MAXLAY
      NELDIM  = NELCMP * NELLAY
C
C ************************************************************************
C
      CALL SEARCI
     1     (INT(IELV),INT(INLV),ISLV,MAXNFR,NNODE,1,NELNOD,NELNFR)
      CALL SEARCH
     1     (REA(ICOR),REA(INOD),ISLV,MAXCRD,NNODE,1,NELNOD,NELCRD)
      CALL SEARCH
     1     (REA(IXRL),REA(IDTOT),INT(IELV),1,NFD,1,NELLV,1)
      CALL SEARCH
     1     (REA(IXIRL),REA(IDINC),INT(IELV),1,NFD,1,NELLV,1)
      CALL SEARCH
     1     (REA(KDMTNO),REA(IDMTNO),ISLV,MAXST2,NNODE,1,NELNOD,NELST2)
      CALL SEARCH
     1     (REA(KGEPNO),REA(IGEPNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1     (REA(KIGENO),REA(IIGENO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1     (REA(KGSINO),REA(IGSINO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1     (REA(KIGSNO),REA(IIGSNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1     (REA(KGTDNO),REA(IGTDNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
```

```
     1      (REA(KTDSNO),REA(ITDSNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1      (REA(KEQPSI),REA(IEQPSI),ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KEQPST),REA(IEQPST),ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KIPSNO),REA(IIPSNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1      (REA(KICSNO),REA(IICSNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1      (REA(KITSNO),REA(IITSNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1      (REA(KEQCSI),REA(IEQCSI),ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KIOMNO),REA(IIOMNO),ISLV,MAXSTR,NNODE,1,NELNOD,NELSTR)
      CALL SEARCH
     1      (REA(KSWLNO),REA(ISWELL),ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KTMPNO),REA(ITEM)   ,ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KTDFNO),REA(ITEMDF),ISLV,    1   ,NNODE,1,NELNOD,    1  )
      CALL SEARCH
     1      (REA(KISTRS),REA(ICHAR)  ,ISLV,MAXCHR,NNODE,1,NELNOD,NELCHR)
C
                            N3HARD = NHARD * 3
      CALL SEARCH
     1      (REA(KYIELD),REA(IYIELD),ISLV,N3HARD,NNODE,1,NELNOD,N3HARD)
      CALL INTERP
     1      (REA(ICH)     ,REA(KISTRS), 1 ,NELNOD,NELCHR)
C
C     D-MATRIX ROTATION FOR ANISOTROPIC MATERIALS
C
      IF (NONISO .EQ. 1) CALL ROTDMT
     1      (REA(KDMTNO),REA(IPREF ),ISLV   ,NELNOD,NELSTR,NNODE ,
     2       NDI          ,NSHEAR          )
C
C *** DYNAMIC CALCULATIONS: QUANTITIES ASSOCIATED WITH NODAL TIME
C *** DERIVATIVES
C
      IF( IDYNM .EQ. 1 .AND. JDYN .EQ. 1 ) THEN
C
      CALL SEARCH
     1      ( REA(IAELM),REA(IDYNA),INT(IELV),1,NFD,1,NELLV,1 )
      CALL SEARCH
     1      ( REA(IVELM),REA(IDYNV),INT(IELV),1,NFD,1,NELLV,1 )
```

```
C
                                                  END IF
C
C ***********************************************************************
C
      IF( IC.EQ.75 .AND. JINC.LT.JLARGE )          T H E N
C
C ***********************************************************************
C
      DO 90 I = 1 , NELNOD
      CALL NUL   (VTRANS,   4  )
      CALL NUL   (TTRANS,   9  )
      CALL TSH04N(VTRANS,TTRANS,ETRANS,CTRANS,REA(ICOR),I,
     *            MAXCRD,NELNOD,  SIGN,IFLAG)
C
            I1= NELSTR * IDP *(I - 1)
C
      CALL SHTRAN(REA(KGEPNO+I1),VTRANS,ETRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KIGENO+I1),VTRANS,ETRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KGSINO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KIGSNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KGTDNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KTDSNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
C
C ***
      DO 70 K = 1 ,MAXSTR
            K1= NELSTR * IDP * (NELSTR *(I - 1)+(K - 1))
C
      CALL SHTRAN(REA(KDMTNO+K1),VTRANS,TTRANS,SIGN,NELSTR)
   70 CONTINUE
C
            I2= NELSTR * NELSTR * IDP *(I - 1)
C
      CALL TRANSP(REA(KDMTNO+I2),NELSTR)
      CALL TRANSP(ETRANS,3)
      CALL MATINV(ETRANS,3,3,ISW)
C
      DO 80 J = 1 ,MAXSTR
            J1= NELSTR * IDP * (NELSTR *(I - 1)+(J - 1))
      CALL SHTRAN(REA(KDMTNO+J1),VTRANS,ETRANS,SIGN,NELSTR)
   80 CONTINUE
C
   90 CONTINUE
C
C ***********************************************************************
```

```
C
      E L S E   I F ( IC.EQ.98 .AND.
     &                JINC.LT.JLARGE     )      T H E N
C
C ***********************************************************************
C
      CALL TBM02N(CTRANS,REA(ICOR),MAXCRD,NELNOD)
C
                                                        E N D I F
C
C
      IF( JINC.GE.JLARGE )              T H E N
C
C ***********************************************************************
C
C              UPDATE GEOMETRY FOR LARGE DEFORMATION ANALYSIS
C              ----------------------------------------------
C
C      BEGINNING OF
C       INCREMENT (ISTEP = -1):
C
C                       Xi        =  X0 + Ui
C
C      MID-INCREMENT (ISTEP = 0 ):
C
C                       Xi+1/2    =  X0 + Ui + 1/2*(si)*(Ui+1 - Ui)
C
C      END OF INCREMENT ( ISTEP = 1 ):
C
C                       Xi+1      =  X0 + Ui +     (si)*(Ui+1 - Ui)
C
C ***********************************************************************
C
C
                                                 HALF    = 0.5D0
                                                 ONE     = 1.0D0
C
C
C ***********************************************************************
C
C
C ***********************************************************************
C
       IF( NDIMEN.EQ.MAXNFR )        T H E N
C
```

```
C                                       ==============================
C                                       NUMBER OF COORDINATES IS EQUAL
C                                       TO THE NUMBER OF DEGREES-OF-
C                                       FREEDOM (CONTINUUM ELEMENTS)
C                                       ==============================
C
C
C                                       ------------------------------
C                                       COMPUTE GEOMETRY AT THE
C                                       BEGINNING OF THE INCREMENT
C                                       ------------------------------
C
          CALL ADD( REA(ICOR),REA(ICOR),REA(IXRL ),NDIMEN*NELNOD)
C
      IF( ISTEP.EQ.0 )              T H E N
C
C                                       ------------------------------
C                                       COMPUTE HALF OF INCREMENTAL
C                                       DISPLACEMENTS TO OBTAIN
C                                       MID-INCREMENT GEOMETRY
C                                       ------------------------------
          CALL ADDSMU( REA(ICOR),ONE,REA(ICOR),HALF,REA(IXIRL),
     &                 NDIMEN*NELNOD)
      E L S E  I F( ISTEP.EQ.1 )    T H E N
C                                       ------------------------------
C                                       USE TOTAL INCREMENTAL
C                                       DISPLACEMENTS TO OBTAIN END OF
C                                       INCREMENT GEOMETRY
C                                       ------------------------------
          CALL ADD   ( REA(ICOR),REA(ICOR),REA(IXIRL),NDIMEN*NELNOD)
C
      E N D I F
C
C
        ELSE IF( MAXNFR.GT.NDIMEN )  T H E N
C
C                                       ==============================
C                                       NUMBER OF DEGREES-OF-FREEDOM
C                                       IS GREATER THAN THE NUMBER OF
C                                       COORDINATES (BEAM AND SHELL
C                                       ELEMENTS)
C                                       ==============================
          DO 10000 I=1,NELNOD
C
            IICOR   = ICOR   + (I-1)*NELCRD*IDP
```

```
            IIXRL  = IXRL   + (I-1)*MAXNFR*IDP
            IIXIRL = IXIRL  + (I-1)*MAXNFR*IDP
C
C
C                               ------------------------------
C                               COMPUTE GEOMETRY AT THE
C                               BEGINNING OF THE INCREMENT
C                               ------------------------------
C
            CALL ADD( REA(IICOR),REA(IICOR),
     &                REA(IIXRL),NDIMEN     )
C
            IF( ISTEP.EQ.0 )               T H E N
C
C
C                               ------------------------------
C                               COMPUTE HALF OF INCREMENTAL
C                               DISPLACEMENTS TO OBTAIN
C                               MID-INCREMENT GEOMETRY
C                               ------------------------------
C
            CALL ADDSMU( REA(IICOR),ONE ,REA(IICOR),
     &                           HALF,REA(IIXIRL),NDIMEN)
C
            E L S E  I F( ISTEP.EQ.1 )       T H E N
C
C
C                               ------------------------------
C                               USE TOTAL INCREMENTAL
C                               DISPLACEMENTS TO OBTAIN END OF
C                               INCREMENT GEOMETRY
C                               ------------------------------
            CALL ADD( REA(IICOR),REA(IICOR),
     &                REA(IIXIRL)          ,NDIMEN )
C
            E N D I F
C
10000     CONTINUE
C
      IF( IC.EQ.75 )            T H E N
C
C
C                               ==============================
C                               UPDATE TRANSFORMATIONS
C                               ==============================
C
      DO 20000 I = 1 , NELNOD
C
      CALL NUL   (VTRANS,    4   )
```

```
      CALL NUL   (TTRANS,   9  )
      CALL TSH04N(VTRANS,TTRANS,ETRANS,CTRANS,REA(ICOR),I,
     &              MAXCRD,NELNOD,  SIGN,IFLAG)
C
             I1= NELSTR * IDP *(I - 1)
C
      CALL SHTRAN(REA(KGEPNO+I1),VTRANS,ETRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KIGENO+I1),VTRANS,ETRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KGSINO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KIGSNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KGTDNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
      CALL SHTRAN(REA(KTDSNO+I1),VTRANS,TTRANS,SIGN,NELSTR)
C
      DO 15000 K = 1 ,MAXSTR
            K1= NELSTR * IDP * (NELSTR *(I - 1)+(K - 1))
      CALL SHTRAN(REA(KDMTNO+K1),VTRANS,TTRANS,SIGN,NELSTR)
15000 CONTINUE
C
             I2= NELSTR * NELSTR * IDP *(I - 1)
C
      CALL TRANSP(REA(KDMTNO+I2),NELSTR)
      CALL TRANSP(ETRANS,3)
      CALL MATINV(ETRANS,3,3,ISW)
C
      DO 16000 J = 1 ,MAXSTR
            J1= NELSTR * IDP * (NELSTR *(I - 1)+(J - 1))
      CALL SHTRAN(REA(KDMTNO+J1),VTRANS,ETRANS,SIGN,NELSTR)
16000 CONTINUE
C
20000  CONTINUE
C
      ELSE IF( IC.EQ.98 )                       T H E N
C
C                                ==============================
C                                UPDATE BEAM TRANSFORMATION
C                                ==============================
C
      CALL TBM02N( CTRANS,REA(ICOR),MAXCRD,NELNOD )
C
                                                E N D I F
C
        ELSE IF( NDIMEN.GT.MAXNFR )  T H E N
C
C                                ==============================
C                                SOMETHING IS WRONG!!!!!!!!!
```

```
C                           ============================
          CALL QUIT( 'NO. ','DIME','NS .','GT. ','NO. ','DOF ', 0 )
C
                                    E N D I F
                                    E N D I F
C
C *******************************************************************
C
      RETURN
      END
```

Note that the coordinate transformations necessary for the shell elements take place in this subprogram.

The following subroutine INITI1 calculates pointers for the finite element database:

```
C=SUBROUTINE= INITI1  CALLED BY SUBROUTINE 'BULKIN'
      SUBROUTINE INITI1
     1    (RWORK ,IWORK ,ISIZE )
C
C *******************************************************************
C
C     ALLOCATES CORE FOR DATA INPUT AND ZEROES OUT CORE
C
C *******************************************************************
C
C   VARIABLES
C   =========
C     IAELM    /START6/  ELEMENT ARRAY FOR NODAL ACCELERATION
C     IBASE    /HARMON/  HARMONIC BASE MOTION MAGNITUDE AND PHASE
C     IBETA    /START7/  ELEMENT BETA MATRIX
C     IBSECT   /BSECT/   POINTER TO BEAM SECTION PROPERTY SETS
C     IBTLC    /START5/  POINTER FOR THE LAST ADDRESS
C     ICH      /START6/  ELEMENT MATERIAL PROPERTY ARRAY (LOCAL)
C     ICHAR    /START1/  ELEMENT MATERIAL PROPERTY ARRAY (GLOBAL)
C     ICMFOR   /HARMON/  POINTER TO THE COMPLEX MODAL FORCE
C     ICHHFN   /HARMON/  POINTER TO THE COMPLEX MODAL H-FUNCTION
C     ICMRES   /HARMON/  POINTER TO THE COMPLEX MODAL RESPONSE
C     ICNFOR   /HARMON/  POINTER TO THE COMPLEX NODAL FORCE VECTOR
C     ICOLPT   /START5/  POINTER TO ARRAY DEFINING THE LOCATIONS OF THE
C                        COLUMN ELEMENTS THAT ARE JUST ABOVE THE DIAGONAL
C                        POSITIONS, FOR A GLOBAL ARRAY STORED IN PROFILE FORM
C     ICON     /START7/  ELEMENT CONNECTIVITY ARRAY
C     ICOP     /START1/
```

```
C      ICOR     /START6/  NODAL COORDINATE ARRAY
C      ICSTNO   /START4/  NODAL CREEP STRAIN
C      IDASH    /ADDVAL/  POINTER TO THE ADDED DASHPOT DAMPING VALUES
C      IDET     /START7/  DETERMINANTS ARRAY AT INTEGRATION POINTS
C      IDFINC   /START4/  INCREMENTAL DEFORMATION GRADIENT ARRAY
C      IDFTOT   /START4/  TOTAL DEFORMATION GRADIENT ARRAY
C      IDIAG    /START5/  POINTER TO DIAGONAL COMPONENTS OF GLOBAL STIFFNESS
C                         ARRAY STORED IN PROFILE FORM
C      IDINC    /START4/  INCREMENTAL DISPLACEMENT ARRAY
C      IDINCP   /START5/  ITERATION  VECTOR FOR INCREMENTAL DISPLACEMENT
C      IDIST    /START1/  DISTRIBUTED LOAD ARRAY
C      IDMTNO   /START4/  NODAL DMATRIX ARRAY ALLOCATION
C      IDP      /MACHIN/  WORD LENGTH ( 1 REAL WORD = IDP INTEGER WORD )
C      IDSITR   /START4/  DISTRIBUTED LOAD INPUT ARRAY
C      IDSX1    /START4/  WORKING ARRAY FOR DYNAMICS
C      IDSX2    /START4/  WORKING ARRAY FOR DYNAMICS
C      IDTOT    /START4/  TOTAL DISPLACEMENT VECTOR
C      IDUP     /START2/  DUPLICATED NODE CONNECTIVITY
C      IDYNA    /START4/  GLOBAL ACCELERATION ARRAY
C      IEQCST   /START4/
C      IEQPSI   /START6/  INCREMENTAL NODAL EQUIVALENT PLASTIC
C                         STRAIN BY LAYER
C      IEQPST   /START6/  TOTAL NODAL EQUIVALENT PLASTIC STRAIN BY LAYER
C      IETM     /START6/
C      IEXT     /START3/
C      IFORCE   /START4/
C      IFORIN   /START5/
C      IFREQ    /SHIFT /  HIGH FREQUENCY BOUNDS FOR POWER SHIFT
C      IGEPNO   /START8/  TOTAL GENERALIZED NODAL STRAIN COMPONENTS
C      IGNMS    /EIGEN/
C      IGSTNO   /START8/  TOTAL GENERALIZED NODAL STRESS COMPONENTS
C      IGTDNO   /START8/
C      IHARM    /HARMON/  HARMONIC NODAL FORCE MAGNITUDE AND PHASE
C      IICSNO   /START4/
C      IIGENO   /START8/  INCREMENTAL GENERALIZED NODAL STRAIN
C                         COMPONENTS
C      IIGSNO   /START8/  INCREMENTAL GENERALIZED NODAL STRESS
C                         COMPONENTS
C      IIOMNO   /START4/  INCREMENTAL NODAL BACKSTRESS (SHIFT TENSOR)
C                         COMPONENTS BY LAYER
C      IIPSNO   /START4/  INCREMENTAL NODAL PLASTIC STRAIN COMPONENTS
C                         BY LAYER
C      IISNNO   /START4/  INCREMENTAL NODAL STRAIN COMPONENTS
C                         FOR SHELL ELEMENTS BY LAYER
C      IISTNO   /START4/  INCREMENTAL NODAL STRESS COMPONENTS
```

```
C                          FOR SHELL ELEMENTS BY LAYER
C     IITSNO  /START4/     INCREMENTAL THERMAL STRAIN AT NODES
C     IKBC    /START3/     POINTER FOR THE BOUNDARY CONSTRAINTS
C     IKBCR   /START7/     POINTER FOR THE BOUNDARY CONSTRAINT BUFFER
C     ILAST   /START5/     LAST ADRESS USED IN THE WORKING AREA
C     ILPRNT  /ALGEM/      LINEPRINTER ID.NUMBER
C     IMASDI  /START5/     POINTER TO DIAGONAL COMPONENTS OF GLOBAL
C                          CONSISTENT MASS ARRAY STORED IN PROFILE FORM
C     IMASNO  /START6/     NODAL ARRAY FOR LUMPED MASS MATRIX
C     IMASS   /ADDVAL/     POINTER TO THE ADDED MASS VALUES
C     IMASUP  /START5/     POINTER TO UPPER TRIANGULAR PART OF GLOBAL
C                          CONSISTENT MASS ARRAY STORED IN PROFILE FORM
C     INDISP  /PERIOD/     PERIODIC DISPLACEMENT INPUT ARRAY
C     INEL    /START1/     ELEMENT CONNECTIVITY
C     INFORC  /PERIOD/     PERIODIC NODAL FORCE INPUT ARRAY
C     INLV    /START2/
C     INOD    /START2/
C     IOMEG   /EIGEN/
C     IOMENO  /START4/     TOTAL NODAL BACKSTRESS (SHIFT TENSOR) COMPONENTS
C                          BY LAYER
C     IPDISP  /PERIOD/     PERIODIC DISPLACEMENT PERIOD ARRAY
C     IPFORC  /PERIOD/     PERIODIC NODAL FORCE PERIOD ARRAY
C     IPOST   /START1/     FLAG FOR POST FILE GENERATION
C     IPP     /START6/
C     IPRES   /START1/     FLAG FOR PRESSURE LOADING
C     IPRINT  /START1/     FLAG FOR PRINT OPTION BUFFER
C     IPSTNO  /START4/     TOTAL NODAL PLASTIC STRAIN COMPONENTS
C                          BY LAYER
C     IREAC   /START5/     NODAL EACTION FORCE ARRAY
C     IRESID  /START4/     NODAL RESIDUAL FORCE ARRAY
C     IRL     /START5/
C     IRLB    /START5/
C     ISBC    /START3/     STRESS BOUNDARY CONDITION INDEX ARRAY
C     ISBCR   /START3/     STRESS BOUNDARY CONDITION INPUT BUFFER
C     ISHIFT  /SHIFT /     POINTER FOR THE LIST OF POWER SHIFT POINTS
C     ISIG    /START6/     NODAL TOTAL STRESS ARRAY
C     ISIGNO  /START4/     TOTAL NODAL STRESS COMPONENTS FOR
C                          SHELL ELEMENTS BY LAYER
C     ISIZE   ARGUMENT     SIZE PARAMETER FOR THE CURRENT WORKING STORAGE
C     ISNOD   /START6/
C     ISPRI   /ADDVAL/     POINTER TO THE ADDED SPRING STIFFNESSES
C     ISWELL  /START4/
C     ITDSNO  /START4/
C     ITEM    /START2/
C     ITEMDF  /START2/
```

```
C      ITI     /START3/
C      ITR     /START3/
C      ITRAC   /START3/  POINTER FOR NODAL FORCE INTEGER VALUES
C      ITRACR  /START7/  POINTER FOR NODAL FORCE REAL VALUES
C      ITRAN   /START3/  POINTER ARRAY FOR NODAL TRANSFORMATION
C      ITRANR  /START7/  NODAL COORDNATE TRANSFORMATION BUFFER
C      ITSTNO  /START4/  THERMAL STRAIN ( TOTAL ) AT NODES
C      ITYP    /START1/
C      IUPTRI  /START5/  POINTER TO UPPER TRIANGULAR PART OF GLOBAL
C                        STIFFNESS ARRAY STORED IN PROFILE FORM
C      IVELM   /START6/
C      IVSWTO  /START4/
C      IWNOD   /START6/  NO LONGER USED IN VERSION 1.7 OR UP
C      IWORK   ARGUMENT
C      IWTNOD  /START4/
C      IXC     /START6/
C      IXIRL   /START6/
C      IXK     /START6/
C      IXM     /START6/
C      IXP     /START6/
C      IXRL    /START6/
C      JPEROD  /PERIOD/
C      JDIST   /CONTRO/
C      JSUB    /CONTRO/
C      KBASE   /HARMON/  HARMONIC BASE MOTION NODE AND D.O.F. LIST
C      KBSECT  /BSECT/   POINTER TO NODAL LIST OF BEAM SECTIONS
C      KCSTNO  /START6/
C      KCSTRN  /START6/
C      KDASH   /ADDVAL/  POINTER TO THE LIST OF D.O.F. WITH ADDED DAMPING
C      KDMAT   /START6/
C      KDMTNO  /START6/
C      KEQPSI  /START6/
C      KEQPST  /START6/
C      KGEPNO  /START8/
C      KGEPS   /START8/
C      KGSIG   /START8/
C      KGSINO  /START8/
C      KGTDNO  /START8/  ELEMENT WORK AREA FOR GENERALIZED INITIAL STRESS
C      KGTDST  /START8/  GENERALIZED INITIAL STRESS AT INTEGRATION POINTS
C      KHARM   /HARMON/  HARMONIC NODAL LOAD NODE AND D.O.F. LIST
C      KICSNO  /START6/  ELEMENT WORK AREA FOR CREEP STRAIN INCREMENT
C      KICSTR  /START6/  CREEP STRAIN INCREMENT AT INTEGRATION POINTS
C      KIGENO  /START8/  ELEMENT WORK AREA FOR GENERALIZED STRAIN
C      KIGEPS  /START8/  GENERALIZED STRAIN AT INTEGRATION PONTS
C      KIGSIG  /START8/  GENERALIZED STRESS AT INTEGRATION POINTS
```

```
C      KIGSNO   /START8/   ELEMENT WORK AREA FOR GENERALIZED STRESS
C      KIPSNO   /START6/   ELEMENT WORK AREA FOR INCREMENTAL PLASTIC STRAIN
C      KIPSTR   /START6/   INC.PLASTC STRAIN AT INTEGRATION POINTS
C      KISNNO   /START6/   ELEMENT WORK AREA FOR INC.STRAIN
C      KISTNO   /START6/   ELEMENT WORK AREA FOR INC.STRESS
C      KISTRN   /START6/   INC.STRAIN AT INTEGRATION POINTS
C      KISTRS   /START6/   INC.STRESS AT INTEGRATION POINTS
C      KITSNO   /START6/   ELEMENT WORK AREA FOR INC.THERMAL STRAIN
C      KITSTR   /START6/   INC.THERMAL STRAIN AT INTEGRATION POINTS
C      KMASS    /ADDVAL/   POINTER TO THE LIST OF D.O.F. WITH ADDED MASS
C      KPSTNO   /START6/   ELEMENT WORK AREA FOR PLASTIC STRAIN
C      KPSTRN   /START6/   PLASTIC STRAIN AT INTEGRATION POINTS
C      KSHIFT   /SHIFT /   POINTER FOR THE NUMBER OF MODES ON EACH SHIFT
C      KSPRI    /ADDVAL/   POINTER TO THE LIST OF D.O.F. WITH ADDED STIFFNESS
C      KTDSNO   /START6/
C      KTSTNO   /START6/
C      KTSTRN   /START6/
C      LFREQ    /SHIFT /   LOW FREQUENCY BOUNDS FOR POWER SHIFT
C      MAXBET   STACK
C      MAXBSP   /MAXIMA/   MAXIMUM NUMBER OF BEAM SECTION PROPERTIES
C      MAXCHR   /MAXIMA/
C      MAXCMP   /MAXIMA/
C      MAXCRD   /MAXIMA/
C      MAXDMT   STACK
C      MAXINT   /MAXIMA/   MAXIMUM NUMBER OF INTEGRATION POINTS / ELEMENT
C      MAXLAY   /MAXIMA/   MAXIMUM NOMBER OF INTEGRATION LAYERS / ELEMENT
C      MAXNFR   /MAXIMA/   MAXIMUM NUMBER OF NODAL D.O.F. / NODE
C      MAXNLV   /MAXIMA/
C      MAXNOD   /MAXIMA/   MAXIMUM NUMBER OF NODES / ELEMENTS
C      MAXPRS   /MAXIMA/   MAXIMUM NUMBER OF DIST. LOAD ENTRIES / ELEMENT
C      MAXSTR   /MAXIMA/   MAXIMUM NUMBER OF STRESS COMPONENTS / NODE
C      MAXWRK   /MAXIMA/
C      NBASE    /PARAM/    NUMBER OF D.O.F. WITH HARMONIC BASE MOTION IMPOSED
C      NBC      /PARAM/    NUMBER OF BOUNDARY CONDITIONS
C      NBSECT   /PARAM/    NUMBER OF BEAM SECTION PROPERTY SETS
C      NDASH    /PARAM/    NUMBER OF D.O.F. WITH ADDED DAMPING
C      NDUP     /PARAM/    NUMBER OF DUPLICATED NODAL POINTS
C      NELEM    /PARAM/    TOTAL NUMBER OF ELEMENTS
C      NEXT     /PARAM/    NOT USED IN VERSION 2.0
C      NFOUND   /SHIFT/    NUMBER OF EIGENVALUES/EIGENVECTORS FOUND IN P-SHIFT
C      NFRSUB   /SUBSTR/   NOT USED IN VERSION 2.0
C      NHARM    /PARAM/    NUMBER OF D.O.F. WITH HARMONIC NODAL FORCES APPLIED
C      NLVSUB   /SUBSTR/   NOT USED IN VERSION 2.0
C      NMASS    /PARAM/    NUMBER OF D.O.F. WITH ADDED MASS
C      NMAX     /PARAM/
```

```
C      NNODE   /PARAM/   NUMBER OF NODE IN THE MESH
C      NOFFST  /SHIFT/   NUMBER OF OFFSETS FOR NEW EIGENVALUES/EIGENVECTORS
C      NPRINT  /PARAM/   NUMBER OF PRINT OPTIONS
C      NSBC    /PARAM/   NUMBER OF STRESS BOUNDARY CONDITIONS
C      NSBNC   /PARAM/
C      NSHIFT  /PARAM/   NUMBER OF POINTS FOR POWER SHIFT
C      NSPRI   /PARAM/   NUMBER OF D.O.F. WITH ADDED STIFFNESS
C      NSUB    /PARAM/   NOT USED IN VERSION 1.7 AND UP
C      NSUMAX  /MAXIMA/  NOT USED IN VERSION 1.7 AND UP
C      NSUMCH  /PARAM /  SUM OF PROFILE COLUMN HEIGHTS FOR UPPER
C                        TRIANGULAR PART OF GLOBAL ARRAY STORED IN PROFILE FORM
C      NTIE    /PARAM/   NUMBER OF TYING CONSTRAINT DATA SETS
C      NTRAC   /PARAM/   NUMBER OF NODAL FORCE VECTOR INPUT
C      NTRAN   /PARAM/   NUMBER OF NODAL COORDINATE TRANSFORMATION DATA
C      NTYPE   /PARAM/
C      OMEGB   /HARMON/  FREQUENCY OF EXCITATION FOR HARMONIC BASE MOTION
C      OMEGH   /HARMON/  FREQUENCY OF EXCITATION FOR HARMONIC NODAL LOADS
C
C ********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
               REAL*4 RWORK
C
C *** -1- ************************************************************
C
      COMMON / ADDVAL / ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / DAMP   / DAMPF(3)
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / MODSUP / IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      COMMON / HARMON / OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
```

```
     1                ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                ICCMAT
      COMMON / PERPAR / IPTYPE(    32),NPTYPE,NPVARS,NPSETS,JPERT ,     NESSUS
     1                NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012         NESSUS
      COMMON / PERPTR / IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,      NESSUS
     1                IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,        NESSUS
     2                IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK         NESSUS
      COMMON / PERDAT / IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,      NESSUS
     1                IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,        NESSUS
     2                IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND         NESSUS
      COMMON / POWER  / IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                IPSF  ,IPSD
      COMMON / PULSES / IPULSE,KPULSE,IPDTIM,IPDFOR
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
      COMMON / PERIOD / JPEROD(    2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / SUBELM / ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
      COMMON / SHIFT  / ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
      COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
     1                NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
      COMMON / BSECT  / IBSECT,KBSECT
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
```

```
     5                  IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                  IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                  IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                  IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                  IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                  IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                  IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                  IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                  KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                  KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                  IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                  KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                  IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                  IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                  IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                  KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                  KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      COMMON / SUBSTR / NLVSUB(   10),NFRSUB(   10)
      COMMON / MACHIN / IDP
      COMMON / ERRORS / IERR
C
C **********************************************************************
C
      DIMENSION RWORK ( ISIZE ) , IWORK ( ISIZE )
C
      DIMENSION YIELDC(   3   )
C
C *** -2- **************************************************************
C
      MAXCRD = 1
      MAXNFR = 2
      MAXNOD = 1
      MAXSTR = 1
      MAXCHR = 1
      MAXINT = 2
      MAXPRS = 1
      MAXLAY = 1
      MAXBET = 1
      MAXWRK = 1
      MAXDMT = 1
```

```
      MAXNLV = 1
      MAXCMP = 1
      MAXBSP = 6
      MAXLWK = 1
C
      MAXELM = NELEM + NDUP + NTIE
C
      YIELDC(1) = 1.00D+36
      YIELDC(2) = 0.00D+00
      YIELDC(3) = 0.00D+00
C
C
C ********************************************************************
C
      MAXNLV = NSUMAX
      IELPRM = 1
C ***
      IF(NTYPE.EQ.0) GO TO  2
C
C ********************************************************************
C
      DO 1 I = 1 ,NTYPE
          IC = I
         IS1 = IELPRM + (IC-1)*13 + 1
C ***
      IF(MAXCRD.LT.IWORK(IS1)   ) MAXCRD = IWORK(IS1)
      IF(MAXNFR.LT.IWORK(IS1+1)) MAXNFR = IWORK(IS1+1)
      IF(MAXNOD.LT.IWORK(IS1+2)) MAXNOD = IWORK(IS1+2)
      IF(MAXSTR.LT.IWORK(IS1+3)) MAXSTR = IWORK(IS1+3)
      IF(MAXCHR.LT.IWORK(IS1+4)) MAXCHR = IWORK(IS1+4)
      IF(MAXINT.LT.IWORK(IS1+5)) MAXINT = IWORK(IS1+5)
      IF(MAXPRS.LT.IWORK(IS1+6)) MAXPRS = IWORK(IS1+6)
      IF(MAXLAY.LT.IWORK(IS1+7)) MAXLAY = IWORK(IS1+7)
      IF(MAXCMP.LT.IWORK(IS1+8)) MAXCMP = IWORK(IS1+8)
                                 MWRK   = IWORK(IS1+3)*  MAXINT
      IF(MAXWRK.LT.MWRK)         MAXWRK = MWRK
                                 MNLV   = IWORK(IS1+1)*IWORK(IS1+2)
      IF(MAXNLV.LT.MNLV)         MAXNLV = MNLV
                                   MWBE = IWORK(IS1+1)*IWORK(IS1+2)
     1                                  * IWORK(IS1+3)*  MAXINT
      IF(MWBE.GT.MAXBET)         MAXBET = MWBE
                                   MLWK =   MAXINT   *IWORK(IS1+7)
     1                                  * IWORK(IS1+8)
      IF(MAXLWK.LT.MLWK)         MAXLWK = MLWK
                                   MDMT = IWORK(IS1+3)*IWORK(IS1+3)
```

```
     1                                   *   MAXINT
      IF(MDMT.GT.MAXDMT)          MAXDMT = MDMT
C
C ***********************************************************************
C
    1 CONTINUE
    2 CONTINUE
C
                                       MAXTEM = MAXLAY
      ITYP     = ILAST   + 1
      INEL     = ITYP    + MAXELM
      ICON     = INEL    + MAXNOD * MAXELM
      ICOP     = ICON    + MAXNOD * MAXELM
C ***
                            IDIST = ICOP
      IF( JFRONT .NE. 0 ) IDIST = ICOP    + MAXELM
      IF( JFRONT .EQ. 0 ) IDIST = ICOP    + NSUMAX * NSUB * 3
      IF( JDIST  .GT. 0 ) IDIST = IDIST   + MAXELM
                            ILAST = IDIST   + MAXELM
C
C ***********************************************************************
C *** CORE ALLOCATION FOR BEAM SECTION PROPERTY SETS                 ***
C ***********************************************************************
C
      IF ( NBSECT .EQ. 0 ) GO TO 110
C
      IBSECT = ILAST
      KBSECT = IBSECT + NBSECT * MAXBSP * IDP
      ILAST  = KBSECT + NBSECT
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  110 CONTINUE
C
C **********************************************************************
C *** CORE ALLOCATION FOR POWER SHIFT IN EIGEN ANALYSIS             ***
C **********************************************************************
C
      IF ( NSHIFT .EQ. 0 ) GO TO 120
C
      ISHIFT = ILAST
      IFREQ  = ISHIFT + NSHIFT * IDP
      LFREQ  = IFREQ  + NSHIFT * IDP
      KSHIFT = LFREQ  + NSHIFT * IDP
      ILAST  = KSHIFT + NSHIFT
```

```
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  120 CONTINUE
C
C ***********************************************************************
C *** CORE ALLOCATION FOR ADDED STIFFNESS, DAMPING AND MASS          ***
C ***********************************************************************
C
      IF ( NSPRI .EQ. 0 ) GO TO 132
C
      ISPRI  = ILAST
      KSPRI  = ISPRI  + NSPRI  * IDP
      ILAST  = KSPRI  + NSPRI  * 2
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  132 CONTINUE
      IF ( NDASH .EQ. 0 ) GO TO 134
C
      IDASH  = ILAST
      KDASH  = IDASH  + NDASH  * IDP
      ILAST  = KDASH  + NDASH  * 2
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  134 CONTINUE
      IF ( NMASS .EQ. 0 ) GO TO 136
C
      IMASS  = ILAST
      KMASS  = IMASS  + NMASS  * IDP
      ILAST  = KMASS  + NMASS  * 2
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  136 CONTINUE
C
C ***********************************************************************
C *** CORE ALLOCATION FOR PULSE LOAD DEFINITION                      ***
C ***********************************************************************
C
      IF( NPULSE .EQ. 0 ) GO TO 138
C
      IPDTIM = ILAST
      IPDFOR = IPDTIM + NPDPTS * IDP
```

```
      IPULSE = IPDFOR + NPDPTS * IDP
      KPULSE = IPULSE + NPULSE * 2 * IDP
      ILAST  = KPULSE + NPULSE * 2
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
  138 CONTINUE
C
C *********************************************************************
C     CORE ALLOCATION FOR FLAGS AND COUNTERS ASSOCIATED WITH EACH GLOBAL
C     ELEMENT FOR THE TREATMENT OF EMBEDDED SINGULARITIES BY MEANS OF
C     SUBELEMENT MESH REPRESENTATIONS
C *********************************************************************
C
                                IEMBED = ILAST
      IF( JEMBED .GT. 0 )       IEMBED = ILAST  + NELEM
                                ISUBEL = IEMBED
      IF( JEMBED .GT. 0 )       ISUBEL = IEMBED + NELEM
                                ISUBNP = ISUBEL
      IF( JEMBED .GT. 0 )       ISUBNP = ISUBEL + NELEM
                                ISUBPT = ISUBNP
      IF( JEMBED .GT. 0 )       ISUBPT = ISUBNP + NELEM
                                ISUBTY = ISUBPT
                                                         NSDATA = 34
      IF( JEMBED .GT. 0 )       ISUBTY = ISUBPT + NELEM * NSDATA
C
C *********************************************************************
C *** CORE ALLOCATION FOR ELEMENT INTEGRATION POINT WORKING ARRAYS   ****
C *********************************************************************
C
                                   ICHAR = ISUBTY + MAXELM
      IF (ICHAR.EQ.2*(ICHAR/2)) ICHAR = ICHAR  +   1
      IPRES   = ICHAR +     MAXCHR * NNODE * IDP
      ISIG    = IPRES + 2 * MAXPRS * NNODE * IDP
      IEPS    = ISIG   + MAXLWK * IDP
      KPSTRN  = IEPS   + MAXLWK * IDP
      KCSTRN  = KPSTRN + MAXLWK * IDP
      KTSTRN  = KCSTRN + MAXLWK * IDP
      KDMAT   = KTSTRN + MAXLWK * IDP
      KGEPS   = KDMAT  + MAXDMT * IDP
      KIGEPS  = KGEPS  + MAXWRK * IDP
      KGSIG   = KIGEPS + MAXWRK * IDP
      KIGSIG  = KGSIG  + MAXWRK * IDP
      KGTDST  = KIGSIG + MAXWRK * IDP
      KISTRS  = KGTDST + MAXWRK * IDP
```

```
      ILAST   = KISTRS + MAXCHR * IDP * MAXNOD
C
      IPRINT  = ILAST
      IMONIT  = IPRINT + 12 * NPRINT
      IPOST   = IMONIT +  4 * NMONIT
      ILAST   = IPOST
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
C **********************************************************************
C *** CORE ALLOCATION FOR NODAL QUANTITIES                           ***
C **********************************************************************
C
      INOD    = ILAST
      IWTNOD  = INOD   + MAXCRD * NNODE  * IDP
      IBPRES  = IWTNOD +          NNODE  * IDP
      IBNORM  = IBPRES + JPRES  * NNODE  * IDP
      ISIGNO  = IBNORM + JPRES  * NNODE  * IDP * 3
      IEPSNO  = ISIGNO + MAXCMP * NNODE  * IDP * MAXLAY
      IPSTNO  = IEPSNO + MAXCMP * NNODE  * IDP * MAXLAY
      ICSTNO  = IPSTNO + MAXCMP * NNODE  * IDP * MAXLAY
      ITSTNO  = ICSTNO + MAXCMP * NNODE  * IDP * MAXLAY
      IISTNO  = ITSTNO + MAXCMP * NNODE  * IDP * MAXLAY
      IISNNO  = IISTNO + MAXCMP * NNODE  * IDP * MAXLAY
      IIPSNO  = IISNNO + MAXCMP * NNODE  * IDP * MAXLAY
      IICSNO  = IIPSNO + MAXCMP * NNODE  * IDP * MAXLAY
      IITSNO  = IICSNO + MAXCMP * NNODE  * IDP * MAXLAY
      ITDSNO  = IITSNO + MAXCMP * NNODE  * IDP * MAXLAY
      IOMENO  = ITDSNO + MAXCMP * NNODE  * IDP * MAXLAY
      IIOMNO  = IOMENO + MAXCMP * NNODE  * IDP * MAXLAY
      IGEPNO  = IIOMNO + MAXCMP * NNODE  * IDP * MAXLAY
      IIGENO  = IGEPNO + MAXSTR * NNODE  * IDP
      IGSINO  = IIGENO + MAXSTR * NNODE  * IDP
      IIGSNO  = IGSINO + MAXSTR * NNODE  * IDP
      IGTDNO  = IIGSNO + MAXSTR * NNODE  * IDP
      IDMTNO  = IGTDNO + MAXSTR * NNODE  * IDP
      IPREF   = IDMTNO + MAXSTR * MAXSTR * NNODE  * IDP
      IYIELD  = IPREF  + NNODE  *    3   * NONISO * IDP
      IMASNO  = IYIELD + NNODE  *    3   * NHARD  * IDP
C ***
      IEQPSI  = IMASNO +          NNODE  * IDP
      IEQPST  = IEQPSI +          NNODE  * IDP * MAXLAY
      IEQCST  = IEQPST +          NNODE  * IDP * MAXLAY
      IEQCSI  = IEQCST +          NNODE  * IDP * MAXLAY
      IVSWTO  = IEQCSI +          NNODE  * IDP * MAXLAY
      ISWELL  = IVSWTO +          NNODE  * IDP * MAXLAY
```

```
      ITEM    = ISWELL +          NNODE  * IDP * MAXLAY
      ITEMDF  = ITEM   +          NNODE  * IDP * MAXLAY
      INLV    = ITEMDF +          NNODE  * IDP * MAXLAY
C
      IF( JLARGE.NE.999999 )      T H E N
C
C                -------------------------
C                LARGE DEFORMATION ANALYSIS
C                -------------------------
C
                 IDFTOT = INLV   + NNODE*MAXNFR
                 IDFINC = IDFTOT + NNODE*NDIMEN*NDIMEN*IDP
                 ILAST  = IDFINC + NNODE*NDIMEN*NDIMEN*IDP
      E L S E
C                -------------------------
C                SMALL DEFORMATION ANALYSIS
C                -------------------------
C
                 ILAST  = INLV   + NNODE*MAXNFR
C
                                  E N D I F
C
C ***********************************************************************
C *** ALLOCATE STORAGE FOR BOUNDARY CONDITIONS                        ***
C ***********************************************************************
C
      IKBC    = ILAST
      IKBCR   = IKBC   + 3 * NBC
      IF (IKBCR.EQ.2*(IKBCR/2)) IKBCR= IKBCR+1
C
      IEXT    = IKBCR + NBC * IDP
      ITI     = IEXT  + 3   * NEXT
      ITR     = ITI   + 3   * NTIE * (NMAX+1)
      IF (ITR.EQ.2*(ITR/2)) ITR = ITR + 1
C
      ITRAN   = ITR   + NTIE * NMAX * IDP
      ITRANR  = ITRAN + 3   * NTRAN
      IF (ITRANR.EQ.2*(ITRANR/2)) ITRANR= ITRANR+1
C
      ITRAC   = ITRANR + NTRAN * IDP
      ITRACR  = ITRAC  + 4     * NTRAC
      IF (ITRACR.EQ.2*(ITRACR/2)) ITRACR= ITRACR+1
C
      ISBC    = ITRACR + 2 * NTRAC * IDP
      ISBCR   = ISBC   + 2 * NSBC
```

```
      IF(ISBCR.EQ.2*(ISBCR/2)) ISBCR= ISBCR+1
      ILAST   = ISBCR  + NSBC * IDP
C
      IDUP    = ILAST
      ILAST   = IDUP   + 2 * NDUP
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
C *** IF HARMONIC (COMPLEX) NODAL LOADS ARE SPECIFIED, ADD... **********
C
      IHARM  = ILAST
      KHARM  = IHARM  + 2 * NHARM * IDP
      ILAST  = KHARM  + 2 * NHARM
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
C *** IF HARMONIC (COMPLEX) BASE EXCITATIONS ARE SPECIFIED, ADD... *****
C
      IBASE  = ILAST
      KBASE  = IBASE  + 2 * NBASE * IDP
      ILAST  = KBASE  + 2 * NBASE
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
C ***
C
      IGNMS   = ILAST
      IOMEG   = IGNMS  + 2 * NSBNC * IDP
      ILAST   = IOMEG  + 2 * NSBNC * IDP
C
C **********************************************************************
C *** CORE ALLOCATION FOR MODAL SUPERPOSITION ANALYSIS              ***
C **********************************************************************
C
      IF ( LDYN .NE. 2 .AND. NPSETS .EQ. 0 ) GO TO 180
C
      IMDAM   = ILAST
      IOMEGD  = IMDAM  + NDYNMD * IDP
      IMFOR0  = IOMEGD + NDYNMD * IDP
      IMDIS0  = IMFOR0 + NDYNMD * IDP
      IMVEL0  = IMDIS0 + NDYNMD * IDP
      IMFOR1  = IMVEL0 + NDYNMD * IDP
      IMDIS1  = IMFOR1 + NDYNMD * IDP
      IMVEL1  = IMDIS1 + NDYNMD * IDP
      ILAST   = IMVEL1 + NDYNMD * IDP
C
      IF ( NHARM .EQ. 0 .AND. NBASE .EQ. 0 ) GO TO 180
C
```

```
      ICMFOR  = ILAST
      ICMRES  = ICMFOR + 2 * NSUPER * IDP
      ICHHFN  = ICMRES + 2 * NSUPER * IDP
      ICBHFN  = ICHHFN + 2 * NSUPER * IDP
      ICCMAT  = ICBHFN + 2 * NSUPER * IDP
      ICBEXC  = ICCMAT + 2 * NBASE  * IDP * 2 * NBASE * IDP
      ILAST   = ICBEXC + 2 * NBASE  * IDP
C
  180 CONTINUE
C
C ***********************************************************************
C *** CORE ALLOCATION FOR FREQUENCY DOMAIN ANALYSIS                   ***
C ***********************************************************************
C
      IF ( LDYN .NE. 4 ) GO TO 280
C
      IMDAM  = ILAST
      IHFN   = IMDAM  + NSUPER            * IDP
      IHFC   = IHFN   + NSUPER *    2     * IDP
      IPSF   = IHFC   + NSUPER *    2     * IDP
      IPSD   = IPSF   + NPSPTS            * IDP
      IFBP   = IPSD   + NPSPTS            * IDP
      ISPP   = IFBP   + NFDPTS            * IDP
      ISFF   = ISPP   + NSUPER * NSUPER * IDP
      ISQQ   = ISFF   + NSUPER * NSUPER * IDP
      ICQQ   = ISQQ   + NSUPER * NSUPER * IDP *    2
      ISIGMO = ICQQ   + NSUPER * NSUPER * IDP *    2
      IEPSMO = ISIGMO + NSUPER * NNODE  * IDP * MAXCMP * MAXLAY
      IELPHI = IEPSMO + NSUPER * NNODE  * IDP * MAXCMP * MAXLAY
      IELTNM = IELPHI +          MAXNOD * IDP * MAXNFR
      ITNM   = IELTNM +          MAXNOD * IDP
      ILAST  = ITNM   + NSUPER * NNODE  * IDP
C
  280 CONTINUE
C
C ***********************************************************************
C *** CORE ALLOCATION FOR ELEMENT NODE WORKING STORAGE                ***
C ***********************************************************************
C
      ICOLPT  = ILAST
      IELV    = ICOLPT + MAXNFR*NNODE
      ICOR    = IELV   + MAXNLV
      IF (ICOR.EQ.2*(ICOR/2)) ICOR= ICOR+1
C
      IETM    = ICOR   + MAXNOD * MAXCRD * IDP
```

```
      ICH      = IETM    + MAXLAY * MAXNOD * IDP
      IPP      = ICH     + MAXCHR * IDP
      IWTNOD   = IPP     + MAXPRS * IDP
      ISNOD    = IWTNOD  + MAXNFR * NNODE
      IF(ISNOD.EQ.2*(ISNOD/2)) ISNOD = ISNOD + 1
      IENOD    = ISNOD   + MAXCMP * MAXNOD * IDP * MAXLAY
      KPSTNO   = IENOD   + MAXCMP * MAXNOD * IDP * MAXLAY
      KCSTNO   = KPSTNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KTSTNO   = KCSTNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KISTNO   = KTSTNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KISNNO   = KISTNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KIPSNO   = KISNNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KICSNO   = KIPSNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KITSNO   = KICSNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KTDSNO   = KITSNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KEQPST   = KTDSNO  + MAXCMP * MAXNOD * IDP * MAXLAY
      KEQPSI   = KEQPST  +          MAXNOD * IDP * MAXLAY
      KEQCSI   = KEQPSI  +          MAXNOD * IDP * MAXLAY
      KIOMNO   = KEQCSI  +          MAXNOD * IDP * MAXLAY
      KSWLNO   = KIOMNO  + MAXSTR * MAXNOD * IDP
      KTMPNO   = KSWLNO  +          MAXNOD * IDP
      KTDFNO   = KTMPNO  +          MAXNOD * IDP
      KDMTNO   = KTDFNO  +          MAXNOD * IDP
      KGEPNO   = KDMTNO  + MAXSTR * MAXSTR * MAXNOD * IDP
      KIGENO   = KGEPNO  + MAXSTR * MAXNOD * IDP
      KGSINO   = KIGENO  + MAXSTR * MAXNOD * IDP
      KIGSNO   = KGSINO  + MAXSTR * MAXNOD * IDP
      KGTDNO   = KIGSNO  + MAXSTR * MAXNOD * IDP
      KYIELD   = KGTDNO  + MAXSTR * MAXNOD * IDP
      IXRL     = KYIELD  + NHARD  * MAXNOD * IDP * 3
      IXIRL    = IXRL    + MAXNLV * IDP
      IVELM    = IXIRL   + MAXNLV * IDP
      IAELM    = IVELM   + MAXNLV * IDP * IDYNM
      IXP      = IAELM   + MAXNLV * IDP * IDYNM
      IXK      = IXP     + MAXNLV * IDP
      IDET     = IXK     + MAXNLV * MAXNLV * IDP
      IXM      = IDET    + MAXINT * IDP
      IXC      = IXM     + MAXNLV * MAXNLV * IDP * IDYNM
      IBETA    = IXC     + MAXNLV * MAXNLV * IDP * IDYNM
      IDTOT    = IBETA   + MAXBET * IDP
      IDYNA    = IDTOT   + NNODE  * MAXNFR * IDP
      IDYNV    = IDYNA   + NNODE  * MAXNFR * IDP * IDYNM
      IRL      = IDYNV   + NNODE  * MAXNFR * IDP * IDYNM
      ILAST    = IRL
C
```

```
C *** PERIODIC LOADING ARRAYS ****************************************
C
      NPDISP = JPEROD( 1 )
      NPFORC = JPEROD( 2 )
C
      IPDISP = ILAST
      IPFORC = IPDISP  + NPDISP* IDP *IDYNM
      INDISP = IPFORC  + NPFORC* IDP *IDYNM
      INFORC = INDISP  + NPDISP    *IDYNM
      ILAST  = INFORC  + NPFORC    *IDYNM
C
      IF (ILAST.EQ.2*(ILAST/2)) ILAST = ILAST + 1
C
      IF ( NHARM .EQ. 0 .AND. NBASE .EQ. 0 ) GO TO 320
      ICNFOR = ILAST
      ILAST  = ICNFOR + NNODE * MAXNFR * IDP * 2
  320 CONTINUE
C
      IRL     = ILAST
      IFORCE  = IRL    + NNODE  * MAXNFR * IDP * 2
      IF( IFSCNT .NE. 0 ) IFORCE = IFORCE + NNODE * MAXNFR * IDP
      IDINC   = IFORCE + NNODE  * MAXNFR * IDP
      IRESID  = IDINC  + NNODE  * MAXNFR * IDP * 2
      IREAC   = IRESID + NNODE  * MAXNFR * IDP * 2
      IF( IPCONJ .NE. 0 ) IREAC  = IREAC  + NNODE * MAXNFR * IDP
      IF( IFSCNT .NE. 0 ) IREAC  = IREAC  + NNODE * MAXNFR * IDP
      IRLB    = IREAC  + NNODE  * MAXNFR * IDP
      IDINCP  = IRLB   + NNODE  * MAXNFR * IDP
      IFORIN  = IDINCP + NNODE  * MAXNFR * IDP
      IBTLC   = IFORIN + NNODE  * MAXNFR * IDP
      IEGNVC  = IBTLC
      IBTLC   = IEGNVC + NSBNC * NNODE  * MAXNFR * IDP *2
      IDSITR  = IBTLC  + NNODE  * MAXNFR * IDP *IDYNM
      IDSX1   = IDSITR + NNODE  * MAXNFR * IDP *IDYNM
      IDSX2   =  IDSX1 + NNODE  * MAXNFR * IDP
      IDSX3   =  IDSX2 + NNODE  * MAXNFR * IDP *IDYNM
      ILAST   =  IDSX3 + NNODE  * MAXNFR * IDP *IDYNM
C
C ********************************************************************
C *** CORE ALLOCATION FOR PERTURBATION DATA SETS                   ***
C ********************************************************************
C
      ISKIP  = ILAST
      IREDEF = ISKIP  + NPSETS + 1
      ILAST  = IREDEF + NPVARS * IDP
```

```
      IF ( ILAST .EQ. 2*(ILAST /2)) ILAST  = ILAST + 1
C
      IF ( NPSETS .EQ. 0 ) GO TO 680
C
C *** ARRAYS NEEDED IN ANY TYPE OF PERTURBATION ANALYSIS ***************
C
      IMEANS = ILAST
      ISTDEV = IMEANS + NPVARS*IDP
      IPDATA = ISTDEV + NPVARS*IDP
      IVTYPE = IPDATA + NPVARS*NPSETS*IDP
      ILAST  = IVTYPE + NPVARS
      IF ( ILAST .EQ. 2*(ILAST /2)) ILAST  = ILAST + 1
C
C *** ARRAYS USED FOR THE PERTURBED STATIC PROBLEM *********************
C
      IDINC0 = ILAST
      IREAC0 = IDINC0 + NNODE*MAXNFR*IDP
      IRESD0 = IREAC0 + NNODE*MAXNFR*IDP
      ILAST  = IRESD0 + NNODE*MAXNFR*IDP
C
C *** ARRAYS USED FOR THE PERTURBED EIGENVALUE PROBLEM *****************
C
      IDGRP  = ILAST
      ILAST  = IDGRP  + 2*NSBNC
C
C *** PERTURBATION WORKSPACE USED FOR VARIABLE MANIPULATIONS ***********
C
      IPWBEG = ILAST
      IXCOOR = IPWBEG
      IXCHAR = IXCOOR + IPTYPE( 1)*NNODE*MAXCRD*IDP
      IXFORC = IXCHAR + IPTYPE( 2)*NNODE*MAXCHR*IDP
      KXFORC = IXFORC + IPTYPE( 3)*NTRAC*2*IDP
      IXDIST = KXFORC + IPTYPE( 3)*NTRAC*4
      IF (IXDIST .EQ. 2*(IXDIST/2)) IXDIST = IXDIST+1
      KXDIST = IXDIST + IPTYPE( 4)*NNODE*2*MAXPRS*IDP
      IXTEMP = KXDIST + IPTYPE( 4)*NELEM
      IF (IXTEMP .EQ. 2*(IXTEMP/2)) IXTEMP = IXTEMP+1
      JXTEMP = IXTEMP + IPTYPE( 5)*NNODE*MAXLAY*IDP
      IXBEAM = JXTEMP + IPTYPE( 5)*NNODE*MAXLAY*IDP
      IXSPRI = IXBEAM + IPTYPE( 6)*NBSECT*MAXBSP*IDP
      KXSPRI = IXSPRI + IPTYPE( 7)*NSPRI*IDP
      IXFVEC = KXSPRI + IPTYPE( 7)*NSPRI*3
      IF (IXFVEC .EQ. 2*(IXFVEC/2)) IXFVEC = IXFVEC+1
      IXPRES = IXFVEC + IPTYPE( 3)*NNODE*MAXNFR*IDP
      IXPREF = IXPRES + IPTYPE( 8)*NNODE*IDP
```

```
      IPWEND = IXPREF + IPTYPE( 9)*NNODE*3*IDP
      ILAST  = IPWEND
  680 CONTINUE
C
      IBTLC   =  ILAST
      ISTRT2  =  IBTLC
C
C *** MESSAGE OUTPUT AND RETURN TO DATA INPUT ROUTINES *****************
C
      ILAST1   = ILAST + 2 * NNODE + 16
      NPX      = ILAST1 - ITYP + 1
      NSHORT   = ILAST1 - ISIZE
C
      CALL LINES(4,4)
      WRITE(ILPRNT,2001) ILAST1
 2001 FORMAT(//,2X,47HNUMBER OF WORDS NEEDED IN BLANK COMMON FOR DATA,
     *6H INPUT,I8,/,2X,43H*******************************************,
     *18H******************)
                                    IF(NSHORT .GT. 0) GO TO  900
C
C **********************************************************************
C **  INITIALIZE DATA IN BLANK COMMON AND CONTINUE EXECUTION          **
C **********************************************************************
C
      CALL NULINT( IWORK(ITYP  ), NPX )
C
C     SPECIAL INITIALIZATION FOR YIELD FUNCTION ARRAY
C
      ISTART = IYIELD
      IEND   = IYIELD + NNODE * 3 * NHARD * IDP
      ISTEP  =                  3 * NHARD * IDP
C
      DO 800 INDEX = ISTART, IEND, ISTEP
        CALL COPY( YIELDC, RWORK(INDEX), 3 )
  800 CONTINUE
C
      RETURN
C
C **********************************************************************
C **  PRINT-OUT THE AMOUNT OF CORE REQUIRED AND TERMINATE EXECUTION  **
C **********************************************************************
C
  900 CONTINUE
               IERR  =  IERR  +  1
                                         CALL LINES( 3 , 3 )
```

```
      WRITE(ILPRNT,2005) NSHORT
 2005 FORMAT(//,2X,'INCREASE BLANK COMMON AND ISIZE WITH',I8,' WORDS')
C
      CALL QUIT('MEMO','RY A','LLOC','ATIO','N   ','    ',IERR)
C
C **********************************************************************
C
      RETURN
      END
```

The following SUBROUTINE INIT2 calculates pointers for the global arrays:

```
C=SUBROUTINE=INITI2  CALLED BY SUBROUTINE 'STRUCT'
      SUBROUTINE INITI2(RWORK,IWORK,ISIZE)
C
C **********************************************************************
C
C     ALLOCATES CORE FOR GLOBAL STIFFNESS
C
C **********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
              REAL*4 RWORK
C
      DIMENSION RWORK(ISIZE),IWORK(ISIZE)
C
C **********************************************************************
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / PERPAR / IPTYPE(   32),NPTYPE,NPVARS,NPSETS,JPERT ,      NESSUS
     1                  NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012       NESSUS
```

```
      COMMON / PERPTR / IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,      NESSUS
     1                  IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,      NESSUS
     2                  IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK       NESSUS
      COMMON / PERIOD / JPEROD(   2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / ERRORS / IERR
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                  IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                  ISBC  ,ISBCR
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                  IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                  IOP   ,IDAMMT,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                  IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                  IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                  IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                  IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                  IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                  IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                  IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                  IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / MACHIN / IDP
C
C *** CONSTRUCT THE REVERSE CONNECTIVITY TABLE *************************
C
      IBEG   = INEL
      IEND   = IBEG   + NELEM*MAXNOD
C
C ... FIND THE MAXIMUM NUMBER OF ELEMENTS AT A NODE
```

```
C
      MAXEAN = 0
      DO 130 NN = 1, NNODE
        NF = 0
        DO 120 IN = IBEG, IEND
          IF ( IWORK(IN) .EQ. NN ) NF = NF+1
  120   CONTINUE
        IF ( MAXEAN .LT. NF ) MAXEAN = NF
  130 CONTINUE
C
C ... ALLOCATE SPACE FOR THE REVERSE CONNECTIVITY TABLE
C
      ILEAN  = ILAST
      ILAST  = ILEAN  + NNODE*MAXEAN
C
      CALL NULINT( IWORK(ILEAN), MAXEAN*NNODE )
C
C ... CONSTRUCT THE REVERSE CONNECTIVITY TABLE
C
      DO 160 NN = 1, NNODE
        NF = 0
        IN = INEL-1
        DO 150 IE = 1, NELEM
          DO 140 IJ = 1, MAXNOD
            IN = IN+1
            IF ( IWORK(IN) .EQ. NN ) THEN
              NF = NF+1
              IM = ILEAN+(NN-1)*MAXEAN+NF-1
              IWORK(IM) = IE
            ENDIF
  140     CONTINUE
  150   CONTINUE
  160 CONTINUE
C
C
C **********************************************************************
C
      IADRES  = ILAST
      IBTLC   = ILAST
      ISKM    = IBTLC  + NEXT*NFD*IDP
      IBQM    = ISKM   + NEXT*NEXT*IDP
      ISRL    = IBQM   + NEXT*IDP
      IAB     = ISRL   + NEXT*IDP
      ILAST   = IAB    + (4*NBAND + 2)*IDP
C
```

```
      IF( MOD( ILAST , 2 ) .EQ. 0 ) ILAST = ILAST + 1
C
C
C *********************************************************************
C
                                    IDIAG  = ILAST
      IF ( IDIAG.EQ.2*(IDIAG/2) ) IDIAG  = IDIAG + 1
                                    IUPTRI = IDIAG + NFD*IDP
C
      IH       = ( NFD + NSUMCH ) * IDP
      NSIZE    =   NFD + NSUMCH
C
      ILAST    = ILAST + IH
                              IOP = ILAST
      IF(MOD(IOP,2) .EQ. 0) IOP = IOP + 1
C
      IF( JDYN .EQ. 1 )  T H E N
C
                         IESIZE = ( MAXNFR * MAXNOD )**2 * IDP
                         IDAMMT = IOP
                         IMASMT = IDAMMT + IESIZE
                         ILAST  = IMASMT + IESIZE
      E N D I F
C
      IF( JEIGEN .EQ. 1 )  T H E N
C
                           IMASDI = IOP
                           IMASUP = IMASDI +         NFD * IDP
                           IPTAR  = IMASDI + IH
                           IPTBR  = IPTAR  + NSBNC * NFD * IDP
                           IPTVED = IPTBR  + NSBNC * NFD * IDP
                           ILAST  = IPTVED + NSBNC * NFD * IDP
      E N D I F
C
      IF( IFBFGS .EQ. 1 )  T H E N
C
                           IPTAR  = IOP
                           IPTBR  = IPTAR  + NFD * NSBFGS * IDP
                           ILAST  = IPTBR  + NFD * NSBFGS * IDP
      E N D I F
C
C *** ADDITIONAL ALLOCATION FOR THE EIGENPROBLEM PERTURBATION **********
C
      IF ( JEIGEN .EQ. 0 .OR. NPSETS .EQ. 0 ) GO TO 300
C
```

```
      ISTIF0 = ILAST
      IMASS0 = ISTIF0 + IH
      IOMEG0 = IMASS0 + IH
      IOMEGP = IOMEG0 + NDYNMD * IDP
      ILAST  = IOMEGP + NDYNMD * IDP
C
C ... CHECK HOW MANY PERTURBATION TERMS THERE IS ROOM IN CORE FOR
C
      ITERM  = (1+2*NFD)*NDYNMD*IDP
      ILEFT  = ISIZE-ILAST
      IF ( ILEFT  .LE. 0 ) ILEFT  = 0
      MAXORD = ILEFT/ITERM
      IF ( MAXORD .GT. (NITER+2)) MAXORD = NITER+2
      IF ( MAXORD .LE.        2 ) MAXORD = 3
C
C ... IN EITHER CASE GO ON AND ESTIMATE THE MINIMAL REQUIREMENTS
C
      IOMEGK = ILAST
      IETAK  = IOMEGK +     NDYNMD*MAXORD*IDP
      IZETAK = IETAK  + NFD*NDYNMD*MAXORD*IDP
      ILAST  = IZETAK + NFD*NDYNMD*MAXORD*IDP
C
  300 CONTINUE
C
C *** CHECK AGAINST THE SIZE OF BLANK COMMON AVAILABLE *****************
C
      CALL LINES ( 7 , 7 )
      WRITE(ILPRNT,2000) NFD , NBAND
 2000 FORMAT(//,2X,21HDATA ON SYSTEM MATRIX,/,
     *          2X,21H*********************,//,
     *          5X,16HNUMBER OF D.O.F.,I15,/,
     *          5X,16HHALF-BANDWIDTH  ,I15)
C
      CALL LINES ( 4 , 4 )
      ILAST1 = ILAST + 16
      NPX    = ILAST1 - IADRES + 1
      WRITE(ILPRNT,2001) ILAST1
 2001 FORMAT(//,2X,44HTOTAL NUMBER OF WORDS NEEDED IN BLANK COMMON,I8,/,
     *2X,52H****************************************************)
      NSHORT = ILAST1 - ISIZE
      IF ( NSHORT .LE. 0 ) GO TO 200
C
C *** WORKING STORAGE NOT SUFFICIENT !
C
      IERR = IERR + 1
```

```
      CALL LINES ( 3 , 3 )
      WRITE(ILPRNT,2005) NSHORT
 2005 FORMAT(//,2X,35HINCREASE BLANK COMMON AND IBLK WITH,I8,6H WORDS)
      CALL QUIT('MEMO','RY A','LLOC','ATIO','N   ','    ',IERR)
C
C ***
C
  200 CONTINUE
      CALL NULINT ( IWORK(IADRES) , NPX )
      RETURN
      END
```

The following SUBROUTINE INITSE defines the pointers for the subelement data storage:

```
C=SUBROUTINE=INITSE  CALLED BY SUBROUTINES 'SUBEIN','SUBDIV'
      SUBROUTINE INITSE
     1     (IWORK ,RWORK ,ISIZE ,KEMBED,KSUBEL,KSUBNP,KSUBTY,NNODE ,
     2      NELEM ,NHARD ,IERR  ,IPOINT,NSDATA,IELEM ,ILAST ,IDP   )
C
C ***********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C ***********************************************************************
C
      DIMENSION IWORK ( ISIZE) ,RWORK ( ISIZE)
      DIMENSION IPOINT(NSDATA)
C
C ***********************************************************************
C
      COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUCMP,
     1                  NSUINT,NSULV ,NSULAY,NSUNDI,NSUSHR,NSUIDF
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
C
C ***********************************************************************
C
C     FUNCTION
C     --------
C     ALLOCATES CORE STORAGE FOR THE SUBELEMENT MESH  ASSOCIATED
C     WITH THE IELM-TH MASTER ELEMENT
C
```

```
C      ARGUMENTS
C      ---------
C      IWORK        INTEGER WORKING SPACE
C      RWORK        SINGLE PRECISION REAL WORK SPACE
C      ISIZE        TOTAL SIZE OF THE WORKING ARRAY
C      KEMBED        THE FLAGS INDICATING THE PRESENCE OF SUB-
C                   ELEMENT DIVISION
C                   = 1  FOR UNIFORM 2X2 SUBELEMENT GRID
C                   = 10 FOR USER DEFINED SUBELEMENT GRID
C                   < 0  SUBELEMENT MESH DEFINED BUT THE CORE IS NOT
C                        ALLOCATED
C      KSUBEL        NUMBER OF SUBELEMENTS
C      KSUBNP        NUMBER OF NODES IN THE SUBELEMENT NODES
C      KSUBTY        SUBELEMENT TYPE
C      NNODE        NUMBER OF NODES IN THE GLOBAL MESH
C      NELEM        NUMBER OF ELEMENTS IN THE GLOBAL MESH
C      IERR         ERROR COUNTER ( IF IERR .GT. 0 ) EVENTUALL THE JOB
C                   SHOULD QUIT
C      IPOINT       ARRAY FOR SUBELEMENT ARRAY POINTERS
C      NSDATA       NUMER OF ENTRIES PER ELEMENT FOR THE SUBELEMENT
C                   POINTER ARRAY
C      IELEM        COUNTER FOR THE CURRENT ELEMENT
C      ILAST        LAST ADDRESS OF THE WORKING STORAGE ( UPDATED IN
C                   THIS ROUTINE )
C      IDP          RATIO OF THE REAL / INTEGER WORD LENGTH
C
C ***********************************************************************
C
C      POINTERS
C      --------
C      IPOINT( - ,IELEM)      I         ARRAY
C      ------------------------------------------------------------------
C             1               I    ELEMENT CONNECTIVITY
C             2               I    PHYSICAL COORDINATES
C             3               I    ELEMENT COORDINATES
C             4               I    TOTAL NODAL DISPLACEMENT
C             5               I    INCREMENETAL NODAL DISPLACEMENT
C             6               I    CORRECTIONAL NODAL DISPLACEMENT
C             7               I    TOTAL NODAL STRAIN
C             8               I    TOTAL NODAL STRESS
C             9               I    TOTAL NODAL PLASTIC STRAIN
C            10               I    TOTAL NODAL CREEP STRAIN
C            11               I    TOTAL NODAL THERMAL STRAIN
C            12               I    GENERALIZED NODAL STRESS (TOTAL)
C            13               I    GENERALIZED NODAL STRAIN (TOTAL)
```

```
C             14                  I    INCREMENTAL NODAL STRAIN
C             15                  I    INCREMENTAL NODAL STRESS
C             16                  I    INCREMENTAL NODAL PLASTIC STRAIN
C             17                  I    INCREMENTAL NODAL CREEP STRAIN
C             18                  I    INCREMENTAL NODAL THERMAL STRAIN
C             19                  I    GENERALIZED NODAL STRESS (INCREMENTAL)
C             20                  I    GENERALIZED NODAL STRAIN (INCREMENTAL)
C             21                  I    LUMPED MASS COEFFICIENT FOR SMOOTHING
C             22                  I    NODAL MATERIAL TANGENT
C             23                  I    NODAL TEMPERATURE
C             24                  I    TEMPERATURE DIFFERENCE
C             25                  I    EQUIVALENT PLASTIC STRAIN (TOTAL)
C             26                  I    EQUIVALENT PLASTIC STRAIN (INCREMENT)
C             27                  I    EQUIVALENT CREEP STRAIN (TOTAL)
C             28                  I    EQUIVALENT CREEP STRAIN (INCREMENT)
C             29                  I    TOTAL SHIFT TENSOR
C             30                  I    INCREMENTAL SHIFT TENSOR
C             31                  I    TVSEWL
C             32                  I    SWELL
C             33                  I    GENRALIZED INITIAL STRESS
C             34                  I    YIELD FUNCTION DEFINED AT NODES
C                                 I
C **********************************************************************
C
      IF( KEMBED .EQ. 0 ) RETURN
C
      WRITE(ICONSL,2000) IELEM
 2000 FORMAT(' SUBELEMENT CORE ALLOCATION FOR ELEMENT NO.',I5)
C
      CALL SUBELV
     1    (IWORK ,KSUBTY,IERR  )
C
      IPOINT(  1 )  = ILAST
      IPOINT(  2 ) = IPOINT(  1 ) +  NSUNOD * KSUBEL
      IPOINT(  3 ) = IPOINT(  2 ) +  NSUCRD * KSUBNP * IDP
      IPOINT(  4 ) = IPOINT(  3 ) +  NSUCRD * KSUBNP * IDP
      IPOINT(  5 ) = IPOINT(  4 ) +  NSUNFR * KSUBNP * IDP
      IPOINT(  6 ) = IPOINT(  5 ) +  NSUNFR * KSUBNP * IDP
      IPOINT(  7 ) = IPOINT(  6 ) +  NSUNFR * KSUBNP * IDP
      IPOINT(  8 ) = IPOINT(  7 ) +  NSUCMP * KSUBNP * IDP
      IPOINT(  9 ) = IPOINT(  8 ) +  NSUCMP * KSUBNP * IDP
      IPOINT( 10 ) = IPOINT(  9 ) +  NSUCMP * KSUBNP * IDP
      IPOINT( 11 ) = IPOINT( 10 ) +  NSUCMP * KSUBNP * IDP
      IPOINT( 12 ) = IPOINT( 11 ) +  NSUCMP * KSUBNP * IDP
      IPOINT( 13 ) = IPOINT( 12 ) +  NSUSTR * KSUBNP * IDP
```

```
      IPOINT( 14 ) = IPOINT( 13 ) +  NSUSTR * KSUBNP * IDP
      IPOINT( 15 ) = IPOINT( 14 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 16 ) = IPOINT( 15 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 17 ) = IPOINT( 16 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 18 ) = IPOINT( 17 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 19 ) = IPOINT( 18 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 20 ) = IPOINT( 19 ) +  NSUSTR * KSUBNP * IDP
      IPOINT( 21 ) = IPOINT( 20 ) +  NSUSTR * KSUBNP * IDP
      IPOINT( 22 ) = IPOINT( 21 ) +           KSUBNP * IDP
      IPOINT( 23 ) = IPOINT( 22 ) +  NSUSTR * KSUBNP * IDP * NSUSTR
      IPOINT( 24 ) = IPOINT( 23 ) +           KSUBNP * IDP
      IPOINT( 25 ) = IPOINT( 24 ) +           KSUBNP * IDP
      IPOINT( 26 ) = IPOINT( 25 ) +           KSUBNP * IDP
      IPOINT( 27 ) = IPOINT( 26 ) +           KSUBNP * IDP
      IPOINT( 28 ) = IPOINT( 27 ) +           KSUBNP * IDP
      IPOINT( 29 ) = IPOINT( 28 ) +           KSUBNP * IDP
      IPOINT( 30 ) = IPOINT( 29 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 31 ) = IPOINT( 30 ) +  NSUCMP * KSUBNP * IDP * NSULAY
      IPOINT( 32 ) = IPOINT( 31 ) +           KSUBNP * IDP
      IPOINT( 33 ) = IPOINT( 32 ) +           KSUBNP * IDP
      IPOINT( 34 ) = IPOINT( 33 ) +  NSUSTR * KSUBNP * IDP
      KLAST        = IPOINT( 34 ) +  NHARD  * KSUBNP * IDP * 3
C
      CALL NULINT( IWORK( ILAST ) , KLAST - ILAST + 1 )
C
C *** UPDATE THE LAST ADRESS
C
                          ILAST    =    KLAST
C
C ***********************************************************************
C
      RETURN
      END
```

The following SUBROUTINE SUBFEM accesses the arrays defined by the subelement pointer analysis:

```
C=SUBROUTINE=SUBFEM  CALLED BY SUBROUTINE 'ASSEM4'
      SUBROUTINE SUBFEM
     1    (RWORK ,IWORK ,ISIZE ,IPOINT,IELEM ,NSUBEL,NSUBNP,NSUBTY,
     2     NSDATA,KEMBED,WORKSP,ISLV  )
C
C ***********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
```

```
              REAL*4   RWORK
C
C ***********************************************************************
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / ELTYP  / NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELPR ,
     1                  NELINT,NELLV ,NELLAY,NDI   ,NSHEAR,NELCMP
      COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUCMP,
     1                  NSUINT,NSULV ,NSULAY,NSUNDI,NSUSHR,NSUIDF
      COMMON / TRANSF / CTRANS(     9),XJACOB(     9)
      COMMON / LOUBIN / JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
     1                  JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / ERRORS / IERR
      COMMON / MACHIN / IDP
      COMMON / TIME   / TIMINC,TOTINC,RUNTIM
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
```

```
     2                 IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                 ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                 IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                 IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                 IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                 IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                 IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                 IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                 IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                 IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                 IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                 IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                 IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                 IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                 KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                 KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                 IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                 KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                 IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                 IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                 IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                 KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                 KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      COMMON / TOLER  / RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
     1                 ENGTOR,ENGNRM
C
C ***********************************************************************
C
      DIMENSION RWORK ( ISIZE) ,IWORK ( ISIZE) ,ISLV  (    1  )
      DIMENSION NSUBEL( NELEM) ,NSUBNP( NELEM) ,NSUBTY( NELEM)
      DIMENSION WORKSP( ISIZE)
      DIMENSION IPOINT(NSDATA, NELEM)
C
C ***********************************************************************
C
C     FUNCTION
C     --------
```

```
C     COMPLETE MIXED FINITE ELEMENT SOLUTION FOR THE SUBELEMENT MESH
C     SUBDIVISION. RESULTS ARE FED BACK TO THE GLOBAL MESH AS A
C     MODIFIED RESIDUAL FORCE VECTOR, AND NODAL QUANTITIES ASSOCIATED
C     WITH THE SUBELEMENT GRIDS ARE RESTORED FOR FURTHER DATA
C     MANUPULATIONS.
C
C     ARGUMENTS
C     ---------
C     RWORK     SINGLE PRECISON REAL WORKAREA
C     IWORK     INTEGER WORKAREA
C     ISIZE     TOTAL SIZE FOR THE WORKAREA
C     IPOINT    POINTER ARRAY FOR THE SUBELEMENT DATA STORAGE
C     IELEM     INDEX FOOR THE CURRENT 'MASTER7 ELEMENT
C     NSUBEL    ARRAY FOR THE NUMBER OF SUBELEMENT ELEMENTS
C     NSUBNP    ARRAY FOR THE NUMBER OF SUBELEMENT NODES
C     NSUBTY    ARRAY FOR THE SUBELEMENT ELEMENT TYPES
C     NSDATA    NUMBER-OF-ENTRIES/ELEMENT IN IPOINT
C     KEMBED    FLAG FOR THE EMBEDDED SUBELEMENTS
C     WORKSP    DUBLE PRECISION REAL WORKSPACE
C
C     POINTERS
C     --------
C     IPOINT( - ,IELEM)       I        ARRAY
C     -----------------------------------------------------------------
C             1               I    ELEMENT CONNECTIVITY
C             2               I    PHYSICAL COORDINATES
C             3               I    ELEMENT COORDINATES
C             4               I    TOTAL NODAL DISPLACEMENT
C             5               I    INCREMENETAL NODAL DISPLACEMENT
C             6               I    CORRECTIONAL NODAL DISPLACEMENT
C             7               I    TOTAL NODAL STRAIN
C             8               I    TOTAL NODAL STRESS
C             9               I    TOTAL NODAL PLASTIC STRAIN
C            10               I    TOTAL NODAL CREEP STRAIN
C            11               I    TOTAL NODAL THERMAL STRAIN
C            12               I    GENERALIZED NODAL STRESS (TOTAL)
C            13               I    GENERALIZED NODAL STRAIN (TOTAL)
C            14               I    INCREMENTAL NODAL STRAIN
C            15               I    INCREMENTAL NODAL STRESS
C            16               I    INCREMENTAL NODAL PLASTIC STRAIN
C            17               I    INCREMENTAL NODAL CREEP STRAIN
C            18               I    INCREMENTAL NODAL THERMAL STRAIN
C            19               I    GENERALIZED NODAL STRESS (INCREMENTAL)
C            20               I    GENERALIZED NODAL STRAIN (INCREMENTAL)
C            21               I    LUMPED MASS COEFFICIENT FOR SMOOTHING
```

```
C              22                  I   NODAL MATERIAL TANGENT
C              23                  I   NODAL TEMPERATURE
C              24                  I   TEMPERATURE DIFFERENCE
C              25                  I   EQUIVALENT PLASTIC STRAIN (TOTAL)
C              26                  I   EQUIVALENT PLASTIC STRAIN (INCREMENT)
C              27                  I   EQUIVALENT CREEP STRAIN (TOTAL)
C              28                  I   EQUIVALENT CREEP STRAIN (INCREMENT)
C              29                  I   TOTAL SHIFT TENSOR
C              30                  I   INCREMENTAL SHIFT TENSOR
C              31                  I   TVSEWL
C              32                  I   SWELL
C              33                  I   GENRALIZED INITIAL STRESS
C              34                  I   YIELD STRESS DEFINED AT NODES
C                                  I
C ************************************************************************
C
      CALL TIMOUT
     1    ('SUBE','LEME','NT L','OOP ','ENTE','RED ')
C
                            WRITE(ICONSL,2000) IELEM
 2000                       FORMAT(30X,'IELEM =',I5)
C
C ************************************************************************
C
              KSUBEL     =     NSUBEL( IELEM )
              KSUBNP     =     NSUBNP( IELEM )
              KSUBTY     =     NSUBTY( IELEM )
              LBACK      =        0
              JMATRX     =        1
              KLAW       =       JLAW
              ICONVG     =        0
              CTOLER     =     RELERR
C
C ************************************************************************
C     INITIALIZE HERE THE NUMBER OF INTEGRATION POINTS / ELEMENT TO BE
C     ABLE TO USE SOME 'TRICKS' OF 'REDUCED' INTEGRATION AT LATER STAGE
C ************************************************************************
C
                         IGAUS     =     0
      IF(JWEIGH .GE. 2)  IGAUS     =     3
      IF(JWEIGH .GE. 4)  IGAUS     =     2
                         ITRANS    =     0
      IF(JWEIGH .GE. 3)  ITRANS    =     1
                         IGAUS     =     0
                         ITRANS    =     0
```

```
C
C ***********************************************************************
C ***                                                                 ***
C *** SET UP CONTROL VARIABLES ASSOCIATED WITH THE 'VARIATIONAL'      ***
C *** STRAIN RECOVERY AND THE NODAL 'RESIDUAL' CALCULATION FOR THE    ***
C *** AUGMENTED LAGRANGIAN TYPE ITERATION                             ***
C ***                      S.N.  12-09-83/01-31-84/03-21-84           ***
C ***                                                                 ***
C *** JINTER = 1     'REDUCED' INTEGRATION FOR THE RECOVERY PROCESS   ***
C ***        = 2     'FULL' INTEGRATION                               ***
C ***        = 3     'TRAPEZOIDAL' INTEGRATION                        ***
C ***        = 4     'SELECTIVE' GAUSS INTEGRATION                    ***
C ***                                                                 ***
C *** JEXTRA = 1     'FULL' INTEGRATION FOR THE RESIDUALS             ***
C ***        = 2     'REDUCED' INTEGRATION                            ***
C ***                                                                 ***
C ***********************************************************************
C
C     JGAUS: CONTROL VARIABLE FOR THE 'RECOVERY' INTEGRATION
C
                            JGAUS     =     1
      IF ( JINTER .EQ. 2 ) JGAUS     =     0
      IF ( JINTER .EQ. 3 ) JGAUS     =     2
      IF ( JINTER .EQ. 4 ) JGAUS     =     3
                            JTRANS    =     0
      IF ( JWEIGH .EQ. 3 ) JTRANS    =     1
C
C     KGAUS: CONTROL VARIABLE FOR THE 'RESIDUAL' INTEGRATION
C
                            KGAUS     =     0
      IF ( JEXTRA .EQ. 2 ) KGAUS     =     1
      IF ( JWEIGH .GE. 2 ) KGAUS     =     3
                            INTRSD    =     NELINT
      IF ( JEXTRA .EQ. 2 ) INTRSD    =     1
                            KTRANS    =     0
      IF ( JWEIGH .EQ. 3 ) KTRANS    =     1
C
C ***********************************************************************
C     ARRANGE SUBELEMENT POINTERS FOR THE GLOBAL WORKAREA
C ***********************************************************************
C
            J  CONNC    =    IPOINT(  1  ,  IELEM   )
            J  PCORD    =    IPOINT(  2  ,  IELEM   )
            J  ECORD    =    IPOINT(  3  ,  IELEM   )
            J  DISPL    =    IPOINT(  4  ,  IELEM   )
```

```
     J  DISPI    =   IPOINT(  5  ,  IELEM   )
     J  DISPC    =   IPOINT(  6  ,  IELEM   )
     J  STRAN    =   IPOINT(  7  ,  IELEM   )
     J  STRST    =   IPOINT(  8  ,  IELEM   )
     J  TPLAS    =   IPOINT(  9  ,  IELEM   )
     J  CREPT    =   IPOINT( 10  ,  IELEM   )
     J  THERT    =   IPOINT( 11  ,  IELEM   )
     J  GSTRS    =   IPOINT( 12  ,  IELEM   )
     J  GSTRN    =   IPOINT( 13  ,  IELEM   )
     J  STRNI    =   IPOINT( 14  ,  IELEM   )
     J  STRSI    =   IPOINT( 15  ,  IELEM   )
     J  PLASI    =   IPOINT( 16  ,  IELEM   )
     J  CREPI    =   IPOINT( 17  ,  IELEM   )
     J  THERI    =   IPOINT( 18  ,  IELEM   )
     J  GSTRI    =   IPOINT( 19  ,  IELEM   )
     J  GSTNI    =   IPOINT( 20  ,  IELEM   )
     J  LUMPM    =   IPOINT( 21  ,  IELEM   )
     J  DMATX    =   IPOINT( 22  ,  IELEM   )
     J  TEMPR    =   IPOINT( 23  ,  IELEM   )
     J  TMPDF    =   IPOINT( 24  ,  IELEM   )
     J  EQPST    =   IPOINT( 25  ,  IELEM   )
     J  EQPSI    =   IPOINT( 26  ,  IELEM   )
     J  EQCST    =   IPOINT( 27  ,  IELEM   )
     J  EQCSI    =   IPOINT( 28  ,  IELEM   )
     J  OMEGT    =   IPOINT( 29  ,  IELEM   )
     J  OMEGI    =   IPOINT( 30  ,  IELEM   )
     J  TVSWL    =   IPOINT( 31  ,  IELEM   )
     J  SWELL    =   IPOINT( 32  ,  IELEM   )
     J  TDSTR    =   IPOINT( 33  ,  IELEM   )
     J  YIELD    =   IPOINT( 34  ,  IELEM   )
C
C
C *********************************************************************
C     PATCHY CODE FOR POINTERS ASSOCIATED WITH THE ELEMENT BUFFER
C *********************************************************************
C
      KEQCSI     =     KEQPST
      KSWLNO     =     KPSTNO
      KIOMNO     =     KCSTNO
C
C *********************************************************************
C     WORKSPACE ALLOCATION FOR THE CURRENT SUBELEMENT SOLUTION
C *********************************************************************
C
                                  IASSEM   =   1
```

```
      CALL SUBELV
     1     ( IWORK , KSUBTY , IERR )
                                                IASSEM    =    0
      CALL SUBALC
     1     ( WORKSP,ISTIFF,IELSTF,ILOADV,IELOAD,ISIZE ,KSUBNP,KSUBEL,
     2       NSUNFR,NSUNOD,NSUSTR,NSUINT,IEBETA,IELDET,IELCOR,IELDMT,
     3       KLAST ,ILAST ,IDP    ,NSUCRD,IELTDS,IELIDS,IELGTD,JELGTD,
     4       JELDMT,KGSIZE,KESIZE,JELEPS,JELTMP,JTDWRK,JELSTR,IELSTR,
     5       JSUCHR,NSUCHR,JREACT,JRESID)
      CALL SMASTR
     1     ( RWORK ,IWORK ,ISLV   ,CTRANS,IELEM ,IC     ,    0   )
C
      IF( JINC .GE. 1 .AND. JITER .GE. 1 )       GO    TO    7900
C
C ***********************************************************************
C     PHASE 0 : PREPARATION OF LUMPED 'GRAMM' MATRIX FOR STRAIN
C               SMOOTHING
C ***********************************************************************
C
      CALL NUL( RWORK(JLUMPM) , KSUBNP )
C
      DO          4900            ISUBEL       =    1          ,      KSUBEL
C
      CALL SNODEL
     1     (ISUBEL         ,IWORK (JCONNC),RWORK (JPCORD),RWORK (JDMATX),
     2      WORKSP(IELCOR),WORKSP(IELDMT),KSUBEL         ,KSUBNP         ,
     3      NSUNOD         ,NSUSTR         ,NSUCRD         ,NSUNFR         ,
     4      RWORK (JDISPI),RWORK (JDISPL),WORKSP(IELIDS),WORKSP(IELTDS),
     5      KCONNC         ,JCONNC         ,RWORK (JTDSTR),WORKSP(IELGTD),
     6      RWORK (JGSTRI),WORKSP(IELSTR))
      CALL LMPMAS
     1     (RWORK (JLUMPM),WORKSP(IELDET),IWORK (KCONNC),NSUNOD         ,
     2      NSUINT         ,KSUBNP         ,KSUBTY         ,RWORK (ICHAR) ,
     3      WORKSP(IELCOR),NSUCRD         ,ISUBEL         ,KSUBEL         ,
     4      RWORK (JPCORD),NSUCRD         ,CTRANS         ,JGRAM          )
C
 4900                 C O N T I N U E
 7900                                          C O N T I N U E
      CALL NUL
     1     (RWORK(JTDSTR),NSUSTR*KSUBNP)
C
C ***********************************************************************
C
C     MIXED ITERATIVE SOLUTION FOR THE SUBELEMENT GRID
C     ----------------------------------------------
```

```
C  (I) TOTAL NUMBER OF ITERATIONS AND CONVERGENCE TOLERANCE ARE SAME AS
C      THOSE DEFINED FOR THE GLOBAL MESH
C (II) IN CASE OF NO CONVERGENCE ACHIEVED (IF NOT DIVERGE), THIS
C      ROUTINE RETURNS NORMALLY TO THE MAIN FINITE ELEMENT SOLUTION
C      WITH A LITTLE WARNING MESSAGE
C
C ***********************************************************************
C      PHASE 1 : NODAL STRESS / MATERIAL TANGENT CALCULATION FOR THE
C                DISPLACEMENT PRECONDITIONING
C ***********************************************************************
C
                        N3HARD = 3 * NHARD
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KYIELD),
     2      N3HARD         ,NELNOD        ,RWORK (JYIELD))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (IXIRL) ,
     2      NSUNFR         ,NELNOD        ,RWORK (JDISPI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KIGENO),
     2      NSUCMP         ,NELNOD        ,RWORK (JSTRNI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KICSNO),
     2      NSUCMP         ,NELNOD        ,RWORK (JCREPI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KITSNO),
     2      NSUCMP         ,NELNOD        ,RWORK (JTHERI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KISTRS),
     2      NSUCHR         ,NELNOD        ,WORKSP(JSUCHR))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KIGSNO),
     2      NELSTR         ,NELNOD        ,RWORK (JGSTRI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KEQCSI),
     2         1           ,NELNOD        ,RWORK (JEQCSI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KIOMNO),
     2      NELSTR         ,NELNOD        ,RWORK (JOMEGI))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KSWLNO),
     2         1           ,NELNOD        ,RWORK (JSWELL))
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KTMPNO),
     2         1           ,NELNOD        ,RWORK (JTEMPR))
```

```
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD        ,KSUBNP        ,RWORK (KTDFNO),
     2         1          ,NELNOD        ,RWORK (JTMPDF))
      CALL SUBT
     1     (WORKSP(JTDWRK),RWORK (JGSTRI),RWORK (JTDSTR),KSUBNP*NSUSTR )
      CALL STRESS
     1     (RWORK (JSTRAN),RWORK (JSTRNI),RWORK (JSTRST),RWORK (JSTRSI),
     2      RWORK (JCREPT),RWORK (JCREPI),RWORK (JTPLAS),RWORK (JPLASI),
     3      RWORK (JTHERT),RWORK (JTHERI),RWORK (JTEMPR),WORKSP(JSUCHR),
     4      NSUSTR        ,NSULAY        ,NSUCHR        ,NSUNOD        ,
     5      KSUBNP        ,NSUNDI        ,NSUSHR        ,JLAW          ,
     6      RWORK (JEQPST),RWORK (JEQPSI),KLAW          ,RWORK (JDMATX),
     7      JMATRX        ,JTEMP         ,JCREEP        ,JITER         ,
     8      JINC          ,NONISO        ,RWORK (JTMPDF),RWORK (JEQCST),
     9      RWORK (JTVSWL),RWORK (JTDSTR),JCONST        ,RWORK (JOMEGT),
     +      RWORK (JOMEGI),ITHERM        ,TIMINC        ,RUNTIM        ,
     1      RWORK (JGSTRN),RWORK (JGSTNI),RWORK (JGSTRS),RWORK (JGSTRI),
     2      NSULAY        ,NSUCMP        ,LBACK         ,RWORK (JSWELL),
     3      RWORK (JEQCSI),RWORK (JPCORD),NSUCRD        ,JPLAST        ,
     4      ICOMPS        ,RWORK (IPREF ),RWORK (JYIELD),NHARD         ,
     5      JUHOOK        ,   1          ,JWKSLP        )
      CALL ADD
     1     (RWORK (JTDSTR),RWORK (JTDSTR),WORKSP(JTDWRK),NSUSTR*KSUBNP )
C
C ***********************************************************************
C              M A I N   I T E R A T I O N   L O O P
C ***********************************************************************
C
                              ISTART     =      1
                              NSTOPS     =     NITER    +    1
C
      DO         8000         ISITER     =    1     ,     NSTOPS
C
                              MCITER     =    ISITER    -    1
C
C ***********************************************************************
C     FIRST ELEMENT LOOP :   STIFFNESS MATRIX ASSEMBLY
C ***********************************************************************
C
CX                         IF ( ISITER  .GT. 1 )  GO TO  5001
C
      DO         5000         ISUBEL     =     1          ,     KSUBEL
C
                                         IASSEM     =    1
      CALL SUBELV
```

```
     1     ( IWORK , KSUBTY , IERR )
      CALL SNODEL
     1     (ISUBEL        ,IWORK (JCONNC),RWORK (JPCORD),RWORK (JDMATX),
     2      WORKSP(IELCOR),WORKSP(IELDMT),KSUBEL        ,KSUBNP        ,
     3      NSUNOD        ,NSUSTR        ,NSUCRD        ,NSUNFR        ,
     4      RWORK (JDISPI),RWORK (JDISPL),WORKSP(IELIDS),WORKSP(IELTDS),
     5      KCONNC        ,JCONNC        ,RWORK (JTDSTR),WORKSP(IELGTD),
     6      RWORK (JGSTRI),WORKSP(IELSTR))
      CALL SUBDER
     1     (WORKSP(IEBETA),WORKSP(IELDET),WORKSP(IELCOR),WORKSP(IELTDS),
     2      WORKSP(IELIDS),RWORK (ICHAR) ,IGAUS         ,ITRANS        ,
     3      CTRANS        )
      CALL INTERP
     1     (WORKSP(JELGTD),WORKSP(IELGTD),NSUINT        ,NSUNOD        ,
     2      NSUSTR        )
      CALL INTERP
     1     (WORKSP(JELDMT),WORKSP(IELDMT),NSUINT        ,NSUNOD        ,
     2      NSUSTR*NSUSTR )
      IF( ISITER .EQ. 1 ) CALL STIFF
     1     (WORKSP(IELSTF),WORKSP(IELOAD),WORKSP(JELGTD),WORKSP(IEBETA),
     2      WORKSP(IELDET),NSUSTR        ,NSUINT        ,NSULV         ,
     3      NSUIDF        ,MCITER        ,WORKSP(JELDMT),     1        ,
     4         1          ,LBACK         ,    0         ,     0        )
      IF( ISITER .EQ. 1 ) CALL SYSEQN
     1     (WORKSP(ISTIFF),WORKSP(ILOADV),WORKSP(IELSTF),WORKSP(IELOAD),
     2      IWORK (JCONNC),NSUNOD        ,NSUNFR        ,KSUBNP        ,
     3      KSUBEL        ,KGSIZE        ,KESIZE        ,ISUBEL        )
C
 5000                C O N T I N U E
 5001                                              CONTINUE
C
      CALL SUBSOL
     1     (WORKSP(ISTIFF),WORKSP(ILOADV),KGSIZE        ,RWORK (JDISPC),
     2      NSUNFR        ,KSUBNP        ,RWORK (JDISPI),NELNFR        ,
     3      NELNOD        ,NSUNOD        ,KEMBED        ,ISITER        ,
     4      WORKSP(JREACT),KSUBEL        )
      CALL ADD
     1     (RWORK (JDISPI),RWORK (JDISPI),RWORK (JDISPC),NSUNFR*KSUBNP )
*     CALL PJOOP
*    1     (RWORK (JDISPI),'DISINC',NSUNFR*KSUBNP)
*     CALL PJOOP
*    1     (RWORK (JDISPC),'DISCOR',NSUNFR*KSUBNP)
C
C ***********************************************************************
C     PHASE 2 : NODAL STRAIN RECOVERY
```

```
C     SECOND ELEMENT LOOP :
C ************************************************************************
C
      CALL NUL
     1     (RWORK (JGSTNI),NSUSTR*KSUBNP )
C
      DO        5100           ISUBEL      =    1         ,     KSUBEL
C
                                                IASSEM    =    0
      CALL SUBELV
     1     ( IWORK , KSUBTY , IERR )
      CALL SNODEL
     1     (ISUBEL        ,IWORK (JCONNC),RWORK (JPCORD),RWORK (JDMATX),
     2      WORKSP(IELCOR),WORKSP(IELDMT),KSUBEL        ,KSUBNP        ,
     3      NSUNOD        ,NSUSTR        ,NSUCRD        ,NSUNFR        ,
     4      RWORK (JDISPI),RWORK (JDISPL),WORKSP(IELIDS),WORKSP(IELTDS),
     5      KCONNC        ,JCONNC        ,RWORK (JTDSTR),WORKSP(IELGTD),
     6      RWORK (JGSTRI),WORKSP(IELSTR))
      CALL SUBDER
     1     (WORKSP(IEBETA),WORKSP(IELDET),WORKSP(IELCOR),WORKSP(IELTDS),
     2      WORKSP(IELIDS),RWORK (ICHAR) ,JGAUS         ,JTRANS        ,
     3      CTRANS        )
      CALL STRAIN
     1     (RWORK (JGSTNI),WORKSP(JELEPS),WORKSP(IELIDS),WORKSP(IEBETA),
     2      NSUSTR        ,NSUINT        ,NSUNFR        ,NSUNOD        ,
     3      NSUIDF        ,KSUBNP        ,RWORK (JLUMPM),WORKSP(IELDET),
     4      RWORK (KCONNC),JGAUS         ,WORKSP(JELTMP),JTEMP         ,
     5      RWORK (JTHERI),RWORK (ICHAR) ,NSUCHR        ,JLAW          ,
     6      NSUSTR        ,KSUBTY        ,WORKSP(IELCOR),NSUCRD        ,
     7      CTRANS        ,RWORK (JSTRNI),NSULAY        ,NSUCMP        ,
     8      WORKSP        ,WORKSP        ,RWORK (JCREPT),RWORK (JTHERT),
     9      JISTRN        ,JCITER        )
C
 5100              C O N T I N U E
C
C
C ************************************************************************
C
      CALL STRESS
     1     (RWORK (JSTRAN),RWORK (JSTRNI),RWORK (JSTRST),RWORK (JSTRSI),
     2      RWORK (JCREPT),RWORK (JCREPI),RWORK (JTPLAS),RWORK (JPLASI),
     3      RWORK (JTHERT),RWORK (JTHERI),RWORK (JTEMPR),WORKSP(JSUCHR),
     4      NSUSTR        ,NSULAY        ,NSUCHR        ,NSUNOD        ,
     5      KSUBNP        ,NSUNDI        ,NSUSHR        ,JLAW          ,
     6      RWORK (JEQPST),RWORK (JEQPSI),KLAW          ,RWORK (JDMATX),
```


C-2

```
     7      JMATRX        ,JTEMP         ,JCREEP        ,ISITER        ,
     8      JINC          ,NONISO        ,RWORK (JTMPDF),RWORK (JEQCST),
     9      RWORK (JTVSWL),RWORK (JTDSTR),JCONST        ,RWORK (JOMEGT),
     +      RWORK (JOMEGI),ITHERM        ,TIMINC        ,RUNTIM        ,
     1      RWORK (JGSTRN),RWORK (JGSTNI),RWORK (JGSTRS),RWORK (JGSTRI),
     2      NSULAY        ,NSUCMP        ,LBACK         ,RWORK (JSWELL),
     3      RWORK (JEQCSI),RWORK (JPCORD),NSUCRD        ,JPLAST        ,
     4      ICOMPS        ,RWORK (IPREF ),RWORK (JYIELD),NHARD         ,
     5      JUHOOK        ,   0          ,JWKSLP        )
C
      DO        5200         ISUBEL      =     1        ,     KSUBEL
C
      CALL SNODEL
     1     (ISUBEL        ,IWORK (JCONNC),RWORK (JPCORD),RWORK (JDMATX),
     2      WORKSP(IELCOR),WORKSP(IELDMT),KSUBEL        ,KSUBNP        ,
     3      NSUNOD        ,NSUSTR        ,NSUCRD        ,NSUNFR        ,
     4      RWORK (JDISPI),RWORK (JDISPL),WORKSP(IELIDS),WORKSP(IELTDS),
     5      KCONNC        ,JCONNC        ,RWORK (JTDSTR),WORKSP(IELGTD),
     6      RWORK (JGSTRI),WORKSP(IELSTR))
      CALL SUBDER
     1     (WORKSP(IEBETA),WORKSP(IELDET),WORKSP(IELCOR),WORKSP(IELTDS),
     2      WORKSP(IELIDS),RWORK (ICHAR) ,IGAUS         ,ITRANS        ,
     3      CTRANS        )
      CALL INTERP
     1     (WORKSP(JELGTD),WORKSP(IELGTD),NSUINT        ,NSUNOD        ,
     2      NSUSTR        )
      CALL INTERP
     1     (WORKSP(JELSTR),WORKSP(IELSTR),NSUINT        ,NSUNOD        ,
     2      NSUSTR        )
      CALL SUBRES
     1     (ISUBEL        ,WORKSP(ILOADV),WORKSP(JELGTD),WORKSP(JELSTR),
     2      IWORK (KCONNC),NSUNOD        ,NSUSTR        ,NSUNFR        ,
     3      NSUINT        ,WORKSP(IELOAD),WORKSP(IEBETA),KESIZE        ,
     4      KSUBNP        ,RWORK (JDISPI),KGSIZE        ,ICONVG        ,
     5      CTOLER        ,WORKSP(IELDET),IWORK (JCONNC),ILPRNT        ,
     6      ICONSL        ,KSUBEL        ,MCITER        ,ENGTOT        ,
     7      WORKSP(JREACT),WORKSP(JRESID))
C
 5200                  C O N T I N U E
C
      CALL SUBCHK
     1     (RWORK (JDISPC),RWORK (JDISPI),KGSIZE        ,DISTOR        ,
     2      ISITER        ,ICONVG        )
C
C ************************************************************************
```

```
C     CHECK THE DISPLACEMENT CONVERGENCE
C ***********************************************************************
C
      IF( ICONVG .EQ. 1 )                       GO   TO   8100
C
 8000                C  O  N  T  I  N  U  E
C
C ***********************************************************************
C     CALCULATE THE CONTRIBUTION OF SUBELEMENT REFINEMENT IN TERMS OF
C     THE GLOBAL RESIDUAL VECTOR
C ***********************************************************************
C
 8100                                          C O N T I N U E
C
      CALL SUBINT
     1     (RWORK (JECORD),NELCRD         ,KSUBNP         ,RWORK (KIGSNO),
     2      NELSTR        ,NELNOD         ,RWORK (JGSTRI))
      IF( JSUBRE   .EQ.   0   )  CALL SUBGLB
     1     (IC            ,RWORK (IXP)    ,RWORK (JSTRST),RWORK (JSTRSI),
     2      IWORK (JCONNC),NSUNOD         ,NSUSTR         ,NSUNFR         ,
     3      NSUINT        ,IDF            ,WORKSP         ,RWORK (IBETA)  ,
     4      RWORK (JECORD),RWORK (ICOR)   ,RWORK (JLUMPM),NELSTR          ,
     5      NSUCRD        ,NELCRD         ,NELNOD         ,KSUBNP         ,
     6      KSUBEL        ,NFD            ,RWORK (IREAC)  ,IWORK (IELV)   ,
     7      RWORK (ISRL)  ,NEXT           ,INEXT          ,RWORK (ICH)    ,
     8      NELCHR        ,NELNFR         ,RWORK (JGSTRI),WORKSP(JTDWRK),
     9      KEMBED        )
      IF( JSUBRE   .NE.   0   )  CALL SUBGLD
     1     (IC            ,RWORK (IXP)    ,RWORK (JSTRST),RWORK (JSTRSI),
     2      IWORK (JCONNC),NSUNOD         ,NSUSTR         ,NSUNFR         ,
     3      NSUINT        ,IDF            ,RWORK (KIGSNO),RWORK (IBETA)  ,
     4      RWORK (JECORD),RWORK (ICOR)   ,RWORK (JLUMPM),NELSTR          ,
     5      NSUCRD        ,NELCRD         ,NELNOD         ,KSUBNP         ,
     6      KSUBEL        ,NFD            ,RWORK (IREAC)  ,IWORK (IELV)   ,
     7      RWORK (ISRL)  ,NEXT           ,INEXT          ,RWORK (ICH)    ,
     8      NELCHR        ,NELNFR         ,RWORK (IDET)   ,NELINT         ,
     9      RWORK (KIGSIG),INTRSD         ,IC             ,KSUBTY         ,
     +      KEMBED        ,WORKSP         ,RWORK (JGSTRI),WORKSP(JTDWRK))
C
C ***********************************************************************
C
      RETURN
      END
```

2.3 Counters and Pointers

Most of the variables used as either counters or pointers are documented in the Fortran source program. In this section, we discuss the counter and pointer variables stored in the common blocks:

START 1

Stores pointers for storing global data and some of the element data. Variables are:

IELPRM	Pointer to the array storing the element parameters for all element types used in the present mesh.
ITYP	Pointer to the element type identifier array.
INEL	Pointer to the element connectivity array.
ICHAR	Pointer to the nodal material property data array.
IPRES	Flag for pressure loading.
IPRINT	Pointer to the line printer output control data array.
IBPRES	Pointer to the nodal pressure definition array.
IBNORM	Pointer to the nodal array storing the components of the nodal point normal.

START 2

Stores pointers for the global mesh data. Variables are:

INOD	Pointer to the nodal coordinate definition array.
ITEM	Pointer to the nodal total temperature array.
ITEMDF	Pointer to the nodal incremental temperature array.
IDUP	Pointer to the array storing the duplicate node input data.

START 3

Stores the pointers for the global nodal constraint and loading data arrays. Variables are:

IKBC	Starting address for the nodal displacement constrain

array.

ITI Pointer to the integer array for defining the tying equations.

ITR Pointer to the real array for defining the tying equations.

ITRAN Starting address of the array for nodal coordinate transformation input.

ISBC Starting address of the integer array for storing nodal stress boundary condition data input.

ISBCR Pointer to the real array for the prescribed nodal stress values.

START 4

Stores the pointer for nodal data storage. Variables are:

IDINC Pointer to the incremental displacement array.

IDTOT Pointer to the total displacement array.

IFORCE Pointer to the total load vector due to the mechanical loading.

IRESID Pointer to the residual vector.

IWTNOD Not used in version 4.2.

ISIGNO Pointer to the total nodal stress array.

IEPSNO Pointer to the total nodal strain array.

IPSTNO Pointer to the total nodal plastic strain array.

ICSTNO Pointer to the total nodal creep strain array.

ITSTNO Pointer to the total nodal thermal strain array.

IISTNO Pointer to the incremental nodal stress array.

IISNNO Pointer to the incremental nodal strain array.

IIPSNO — Pointer to the incremental nodal plastic strain array.

IICSNO — Pointer to the incremental nodal creep strain array.

IITSNO — Starting address for the incremental thermal strain defined at nodes.

IDMTNO — Starting address for the material tangent arrays defined at nodes. In case of shells, the constitutive resultant array is stored.

IOMENO — Starting address for the total shift tensor defined at nodes. This array is used for the kinematic hardening and the unified viscoplastic constitutive models.

ITDSNO — Starting address for the total initial stress terms due to the initial strain terms defined at nodes.

IDYNA — Starting address for the total nodal acceleration array.

IDYNV — Pointer to the total nodal velocity array. IDYNA and IDYNV are defined only when the transient analysis option is invoked.

IPREF — Pointer to the material orientation vector for anisotropic material response defined at a node.

IDFINC — Pointer to the nodal array for the incremental deformation gradient. This array is allocated and used for finite deformation analysis.

IDFTOT — Pointer to the nodal array for the total deformation gradient.

IYIELD — Pointer to the array used to define nodally the strain hardening slope table.

START 5

Stores pointers for global arrays. Variables are:

IRL — Starting address for a vector storing the nodal loads including the residual.

IREAC — Starting address for a vector storing the nodal reactions.

IES	Starting address for the space of banded global stiffness matrix. Not used in Version 4.2.
IAB	Starting address for the space to pull a row of the global stiffness array.
ILAST	Pointer to the last word used in the blank common workspace.
IDIAG	Pointer to the array for diagonal entries of the global stiffness array stored in profile form.
IUPTRI	Pointer to the array storing the upper triangular part of the global stiffness array in profile form.
ICOLPT	Pointer to the integer array storing the column height for each degree of freedom of the global stiffness matrix in profile form.
IMASDI	Pointer to the space reserved for the diagonal entry of the global mass matrix. This space is allocated only when the modal analysis option is invoked.
IMASUP	Pointer to the upper triangular entries of the mass matrix stored in profile form.

START 6

Stores pointers for the element workspace. Variables are:

IELV	Starting address for the connectivity array of the element being processed.
ICOR	Starting address for the nodal coordinate array for the current element.
ISIG	Starting address for the element total stress array at integration points.
IEPS	Starting address for the element total strain array at integration points.
IXRL	Starting address for the total nodal displacement array defined for the current element.

IXIRL	Starting address for the incremental nodal displacement array.
IXP	Starting address for the workspace to store the element load vector.
IXK	Starting address for the workspace for the element stiffness array.
KPSTNO	Pointer to the nodal plastic strain array for the current element.
KCSTNO	Pointer to the nodal creep strain array for the current element.
KTSTNO	Pointer to the nodal thermal strain array for the current element.
KISTNO	Pointer to the nodal incremental stress for the current element.
KISNNO	Pointer to the nodal incremental strain for the current element.
KIPSNO	Pointer to the nodal incremental plastic strain for the current element.
KICSNO	Pointer to the nodal incremental creep strain for the current element
KITSNO	Pointer to the nodal incremental thermal strain for the current element.
KDMAT	Pointer to the element material tangent array defined at the integration points for the current element.
KDMTNO	Pointer to the nodal material tangent array for the current element.
KTDSNO	Pointer to the nodal array for the initial stress due to initial strains such as thermal and creep effects defined for the current element.
KITDST	Pointer to the element initial stress vector.

START 7

Stores more pointers for global and element arrays. Variables are:

ICON Not used in Version 4.2.

IKBCR Starting address for a double precision real array storing the prescribed displacement values.

ITRANR Starting address for a double precision real array storing the angle of rotation for the user specified nodal coordinate transformations.

ITRACR Starting address for a double precision real array storing the values of user specified nodal concentrated loads.

ITRANR Starting address for a double precision real array storing the values of user specified nodal coordinate transformations.

IBETA Starting address for a double precision real array storing the strain displacement matrix at integration points of the current element.

IDET Starting address for a double precision real array storing the determinant of the Jacobian for isoparametric mapping at integration points of the current element.

START 8

Stores pointers for the resultant quantities. Variables are:

KGEPS Pointer to the strain resultant array at element integration points.

KIGEPS Pointer to the incremental strain resultant array at element integration points.

KGEPS Pointer to the strain resultant array at element integration points.

KIGEPS Pointer to the incremental strain resultant array at element integration points.

KGSIG Pointer to the stress resultant array at element

integration points.

KIGSIG Pointer to the incremental stress resultant array at element integration points.

KGTDST Pointer to the total resultant for initial stress at element integration points.

IGEPNO Pointer to the global array for the strain resultants.

IIGENO Pointer to the global array for the incremental strain resultants.

IGSINO Pointer to the global array for the stress resultants.

IIGSNO Pointer to the global array for the incremental stress resultants.

IGTDNO Pointer to the global array for total resultants of initial stress.

KGEPNO Starting address for the nodal array of strain resultants of the current element.

KIGENO Starting address for the nodal array of incremental strain resultants of the current element.

KGSINO Starting address for the nodal array of stress resultants of the current element.

KIGSNO Starting address for the nodal array of incremental stress resultants of the current element.

KGTDNO Starting address for the nodal array of total resultant for initial stresses of the current element.

START 9

Stores pointers for creep related quantities. Variables are:

KEQCSI Pointer to the equivalent incremental creep strain at nodes of the current element.

KEQCST Pointer to the equivalent total creep strain at nodes of

the current element.

3.0 THE FILE SYSTEM

This chapter is devoted to a discussion of the file system for supporting a series of structural analyses by the MHOST program. In the next section, the overview of this file system is presented followed by a detailed discussion on the user interface (input and output files), restart file, and post-processing data file.

3.1 Overview

The MHOST program is a batch processor and when execution is initiated the user does not need to interact with the program until the Fortran STOP statement is executed (or an execution error is detected by the operating system). Under an interactive operational environment, the code produces a summary of the execution log on the terminal screen. Note that the same information is written on the main output file.

The file system is composed of three parts: the standard input and output files referred to as the user interface; the restart file; and the post processing data file. The main report part is referred to as Fortran unit number ILPRNT (see Block Data Subprogram, common block ALGEM) which consists of the input data echo print (card image print out), the input data as it is interpreted by the MHOST code, and the summary of execution such as the usage of memory and time consumed by CPU. The analysis result part is referred to as Fortran unit number JLPRNT. To make this logical distinction clear, separate header records and page counters are provided to these units.

The user interface files are the standard input file, the log file and the line printer output file. All the instructions to accomplish the analysis task are written in the input file. The MHOST program assumes that the input file is a card image file. That is, the input file is a sequentially accessed, formatted file, and the record length is fixed to 80 bytes. Under most operating systems (perhaps IBM systems are the only exceptions), it is not necessary to declare explicitly the record length. As shown in the next section, records are individually read into the read buffer as 80 separate characters and the user interface routine decodes each line.

The output file is assumed to contain 132 bytes/record which is the standard width for most line printers (as well as other printers driven in the same manner as the line printer). Again, except for IBM systems, this file is opened without explicitly specifying the record length. The line

printer file is logically separated into two parts. As the default setting, the code is delivered with the same Fortran unit number.

The log file is the standard output unit on which the system's messages are printed. This file is defined as Fortran unit ICONSL and is handled in a manner which depends on the computer systems default value for interactive terminal output.

Typically, on PRIME systems, ICONSL is the unit number 1 (one) and under the UNIX environment, unit number 6 (six) is assigned to ICONSL.

The restart file is a sequentially accessed binary file. The record length is fixed to 256 bytes/record. The contents of common block and the work space are written and read by the MHOST code. This file is referred to as Fortran unit number IRSTRT. The unit number 8 (eight) is assigned as default value when the MHOST code is delivered.

The post-processing file is a sequentially written formatted file. A record length of 80 bytes is implicitly assumed but not necessarily declared explicitly when the file is opened (under most operating systems). Except for minor differences in the format to write nodal stresses and strains, construction of this file is compatible with the post-processing file output of the MARC general purpose finite element package version K1. Any commercially available finite element post-processing package can easily be modified to read this file. The MENTAT interactive finite element pre- and post-processing package available from MARC is shipped with the MHOST post-processing interface as a standard feature.

3.2 User Interface

The free format reader is written in Fortran 66 to interpret data coming in through the input data file in a line-by-line manner. Two utilities are available. One is the keyword interpreter SUBROUTINE KEY and another is the free format numeric data reader SUBROUTINE FREFOR:

```
C  SUBROUTINE KEY
      SUBROUTINE KEY(NAME,NOPT,IOPT,NN,IN,IERR)
C
C* * * * * *
C
C     CHECKS STRING FOR KEYWORD
C
C     NAME    KEYWORD NAMES
C     NOPT    NUMBER OF KEYWORDS
C     IOPT    KEYWORD FLAG
```

```
C     NN        INTEGER PARAMETERS
C     IN        NUMBER OF INTEGERS
C     IERR      ERROR FLAG
C
C* * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
      DIMENSION NAME(1),NN(4)
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON/FREE/IA(80),IBEGIN(16),ILENGT(16),NSTRIN,IS,ICOL,NEW
      LOGICAL NEW
    1 CONTINUE
      JKEY=1
      CALL STRING(IERR,JKEY,1)
      IS=IS+1
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
      IF(IL.LT.3) GOTO 99
      IF(IL.GT.4) IL=4
      I1=-4
      IB1=IB-1
      DO 3 I=1,NOPT
      I1=I1+4
      DO 2 J=1,IL
      IF(IA(IB1+J).NE.NAME(I1+J)) GOTO 3
    2 CONTINUE
      IOPT=I
      GOTO 4
    3 CONTINUE
      GOTO 99
    4 CONTINUE
C
      IF(IN.EQ.0) GOTO 100
      DO 5 I=1,IN
      IS=IS+1
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
      CALL DECINT(NNI,IA(IB),IL,IERR)
      NN(I)=NNI
    5 CONTINUE
      GOTO 100
C
```

```
   99 CONTINUE
      IERR=IERR+1
      CALL LINES(1,1)
      L=IB-1+IL
      WRITE(ILPRNT,1001) (IA(I),I=IB,IL)
 1001 FORMAT(2X,11H***ERROR***,5X,25HSTRING IS NOT A KEYWORD: ,78A1)
      GOTO 1
  100 CONTINUE
      RETURN
      END

C SUBROUTINE FREFOR
      SUBROUTINE FREFOR(INTW,REAW,NINT,NREA,NVAR,IERR,JKEY)
C
C* * * * * *
C
C     FREE FORMAT ROUTINE
C
C     INTW     INTEGER WORKSPACE
C     REAW     REAL WORKSPACE
C     NINT     NUMBER OF INTEGERS
C     NREA     NUMBER OF REALS
C     NVAR
C     IERR     ERROR FLAG
C     JKEY
C
C* * * * * *
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 REAW
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON/FREE/IA(80),IBEGIN(16),ILENGT(16),NSTRIN,IS,ICOL,NEW
      DIMENSION INTW(1),REAW(1)
      LOGICAL LREA,NEW
      IC=0
      CALL STRING(IERR,JKEY,1)
      IF(JKEY.EQ.1) GOTO 30
      JKEY=0
C
      IF(NINT.LE.0) GOTO 10
      DO 1 I=1,NINT
      IF(ICOL.LE.76) GOTO 2
```

```
      JKEY=-1
      CALL STRING(IERR,JKEY,1)
      IF(JKEY.EQ.1) GOTO 30
      JKEY=0
    2 CONTINUE
      IC=IC+1
      IS=IS+1
      ICOL=ICOL+5
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
      CALL DECINT(IDUM,IA(IB),IL,IERR)
      INTW(IC)=IDUM
    1 CONTINUE
   10 CONTINUE
C
      IF(NREA.LE.0) GOTO 20
      DO 11 I=1,NREA
      IF(ICOL.LE.71) GOTO 12
      JKEY=-1
      CALL STRING(IERR,JKEY,1)
      IF(JKEY.EQ.1) GOTO 30
      JKEY=0
   12 CONTINUE
      IC=IC+1
      IS=IS+1
      ICOL=ICOL+10
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
C
      REAW(IC)=0.
C
      IF( IL .NE. 0 )CALL DECREA(REAW(IC),IA(IB),IL,IERR)
C
C
   11 CONTINUE
   20 CONTINUE
C
      IF(NVAR.EQ.0) GOTO 30
      LREA=NVAR.GT.0
      NTOT=IABS(NVAR)
      IF(ICOL.LE.76) GOTO 23
      JKEY=-1
      CALL STRING(IERR,JKEY,1)
      IF(JKEY.EQ.1) GOTO 30
      JKEY=0
```

```
   23 CONTINUE
      IS=IS+1
      ICOL=ICOL+5
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
      CALL DECINT(NVAR,IA(IB),IL,IERR)
      IF(NVAR.LE.0) GOTO 30
C
      NTOT=NTOT*NVAR
      DO 21 I=1,NTOT
      IF(ICOL.LE.71.OR.(ICOL.LE.76.AND..NOT.LREA)) GOTO 24
      JKEY=-1
      CALL STRING(IERR,JKEY,1)
      IF(JKEY.EQ.1) GOTO 30
      JKEY=0
   24 CONTINUE
      IC=IC+1
      IS=IS+1
      ICOL=ICOL+5
      IB=IBEGIN(IS)
      IL=ILENGT(IS)
      IF(LREA) GOTO 22
      CALL DECINT(IDUM,IA(IB),IL,IERR)
      INTW(IC)=IDUM
      GOTO 21
   22 CONTINUE
      ICOL=ICOL+5
C
      REAW(IC)=0.
C
      IF( IL .NE. 0 )CALL DECREA(REAW(IC),IA(IB),IL,IERR)
C
   21 CONTINUE
   30 CONTINUE
      RETURN
      END
```

An example of usage of the above utilities is the parameter data reader SUBROUTINE DATIN1:

```
C ... SUBROUTINE DATIN1 ... CALLED BY SUBROUTINE 'H O S T' OR 'F E M'
C
      SUBROUTINE DATIN1
     1    (RWORK ,IWORK ,ISIZE ,VERSNO,MONTH ,JDATE ,NELEM ,NNODE ,
     2     NBC   ,NTIE  ,NMAX  ,NTRAN ,NTRAC ,NPOST ,NLVSUB,NFRSUB,
```

```
     3      NEXT   ,JDYN   ,JTEMP ,NPRINT,JREST ,JINC   ,NINC   ,JLOUB ,
     4      JINTER,JEXTRA,JWEIGH,NSTRBC,NTYPE ,MAXSUB,ILAST ,JSUB   ,
     5      NSUB   ,ISTAT ,IDYNM ,ITEST ,JOPTIM,JCREEP,JDIST ,NONISO,
     6      NDYNMD,IDYNMD,NPOSMD,ITHERM,JCONST,NDUP   ,JREPOT,JTANGE,
     7      JTHERM,DALPHA,DBETA ,DGAMMA,JEIGEN,JFORCE,JUTEMP,JUCOEF,
     8      JDISTS,JUHOOK,JDERIV,JUBOUN,JPEROD,NSBNC ,NCREEP,ATOLER,
     9      BTOLER,CTOLER,JPOST ,INTSTR,JBAND ,JFRONT,JDEFOR,NGMRS ,
     +      JEMBED,NBSECT,JDISP ,NSHIFT,NSUPER,JSUBRE,IFBFGS,NSPRI ,
     1      NDASH ,NMASS ,NSBFGS,IFSCNT,IFLINE,IFPRNT,NHARM ,OMEGH ,
     2      NBASE ,OMEGB ,ICOMPS,NPDPTS,NPULSE,IPCONJ,NSSPTS,NSHOCK,
     3      NPSPTS,NFDPTS,LDYN   ,JFDSXX,JISTIF,JCENTM,NHARD ,JFINIT,
     4      JLARGE,JFOLOW,JWKSLP,JISTRN,JCITER,JHRGLS,NDIMEN,JGRAM ,
     5      JPRES ,NMONIT)
C
C **********************************************************************
C
C     READ THE CONTROL DATA AND PARAMETERS ASSOCIATED WITH THE CORE
C     STORAGE ALLOCATION
C
C **********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C
C **********************************************************************
C
      DIMENSION RWORK ( ISIZE) ,IWORK ( ISIZE)
      DIMENSION NFRSUB(MAXSUB) ,NLVSUB(MAXSUB)
      DIMENSION NAME  ( 4, 72) ,NN    (     6)
      DIMENSION NAME1 ( 4, 34) ,NAME2 ( 4, 36)
      DIMENSION NAME3 ( 4,  2)
      DIMENSION JPEROD(     2)
C
      EQUIVALENCE (NAME( 1, 1),NAME1( 1, 1))
      EQUIVALENCE (NAME( 1,35),NAME2( 1, 1))
      EQUIVALENCE (NAME( 1,71),NAME3( 1, 1))
C
C **********************************************************************
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / COUNT  / LININC,LINTOT,NOECHO
      COMMON / CTITLE / TITLE (   20),IDAT  (    5),IDATE2,ICLOCK,
```

```
     1                  IFCRAY
      COMMON / ERRORS / IERR
      COMMON / FREE   / IA     (    80),IBEGIN(    16),ILENGT(    16),
     1                  NSTRIN,IS      ,ICOL   ,NEW
      LOGICAL NEW
C
C *********************************************************************
C
      DATA NAME1
     *         /1HE,1HL,1HE,1HM, 1HN,1HO,1HD,1HE, 1HB,1HO,1HU,1HN,
     *          1HT,1HY,1HI,1HN, 1HT,1HR,1HA,1HN, 1HF,1HO,1HR,1HC,
     *          1HP,1HO,1HS,1HT, 1HS,1HU,1HB,1HS,
     *          1HE,1HX,1HT,1HE, 1HP,1HR,1HE,1HS, 1HT,1HE,1HM,1HP,
     *          1HP,1HR,1HI,1HN, 1HR,1HE,1HS,1HT, 1HL,1HO,1HU,1HB,
     *          1HS,1HT,1HR,1HE, 1HE,1HN,1HD,1H ,
     *          1HT,1HE,1HS,1HT, 1HD,1HY,1HN,1HA, 1HO,1HP,1HT,1HI,
     *          1HT,1HR,1HA,1HC, 1HC,1HR,1HE,1HE, 1HA,1HN,1HI,1HS,
     *          1HM,1HO,1HD,1HA, 1HB,1HU,1HC,1HK, 1HT,1HH,1HE,1HR,
     *          1HC,1HO,1HN,1HS, 1HD,1HI,1HS,1HT, 1HD,1HU,1HP,1HL,
     *          1HR,1HE,1HP,1HO, 1HT,1HA,1HN,1HG, 1HU,1HT,1HH,1HE,
     *          1HS,1HC,1HH,1HE, 1HU,1HF,1HO,1HR, 1HU,1HT,1HE,1HM/
      DATA NAME2
     *         /1HU,1HC,1HO,1HE, 1HU,1HP,1HR,1HE, 1HU,1HH,1HO,1HO,
     *          1HU,1HD,1HE,1HR, 1HU,1HB,1HO,1HU, 1HP,1HE,1HR,1HI,
     *          1HB,1HA,1HN,1HD, 1HF,1HR,1HO,1HN, 1HD,1HE,1HF,1HO,
     *          1HE,1HM,1HB,1HE, 1HG,1HM,1HR,1HS, 1HB,1HE,1HA,1HM,
     *          1HD,1HI,1HS,1HP, 1HS,1HH,1HI,1HF, 1HB,1HF,1HG,1HS,
     *          1HS,1HP,1HR,1HI, 1HD,1HA,1HS,1HH, 1HM,1HA,1HS,1HS,
     *          1HS,1HE,1HC,1HA, 1HL,1HI,1HN,1HE, 1HH,1HA,1HR,1HM,
     *          1HB,1HA,1HS,1HE, 1HC,1HO,1HM,1HP, 1HP,1HU,1HL,1HS,
     *          1HC,1HO,1HN,1HJ, 1HS,1HH,1HO,1HC, 1HP,1HO,1HW,1HE,
     *          1HN,1HO,1HE,1HC, 1HP,1HE,1HR,1HT, 1HS,1HT,1HI,1HF,
     *          1HC,1HE,1HN,1HT, 1HH,1HA,1HR,1HD, 1HF,1HI,1HN,1HI,
     *          1HL,1HA,1HR,1HG, 1HF,1HO,1HL,1HL, 1HW,1HK,1HS,1HL/
      DATA NAME3
     *         /1HH,1HO,1HU,1HR, 1HM,1HO,1HN,1HI/
C
C *********************************************************************
C
C     PARAMETER DATA OPTIONS
C     ======================
C     1           *ELEM MAXIMUM NUMBER AND THE TYPE OF ELEMENT
C     2           *NODE MAXIMUM NUMBER OF NODES
C     3           *BOUN MAXIMUM NUMBER OF DISPLACEMENT CONSTRAINT
C     4           *TYIN FLAG THE TYING OPTION WITH NUMBER OF TYING
```

```
C                          DEGREE OF FREEDOMS
C     5              *TRAN COORDINATE TRANSFORMATION OPTION FLAGGED WITH
C                          THE NUMBER OF POINTS SUBJECTED TO THIS OPER.
C     6              *FORC MAXIMUM NUMBER OF NODAL FORCE DATA
C     7              *POST FLAG THE POST PROCCESSING TAPE GENERATION
C                          OPTION
C     8              *SUBS FLAG THE SUBSTRUCTURING OPTION WITH THE NUNBER
C                          OF SUBSTRUCTURES
C     9              *EXTE
C    10              *PRES FLAG THE NODAL PRESSURE DEFINITION OPTION
C    11              *TEMP FLAG FOR THERMAL LOADING
C    12              *PRIN FLAG FOR PRINT OUTPUT
C    13              *REST FLAG FOR RESTART RUN
C    14              *LOUB SET UP NUMERICAL INTEGRATION
C    15              *STRE FLAG FOR STRESS BOUNDARY CONDITIONS
C    16              *END  .....OBVIOUS.............
C    17              *TEST (RESERVED)
C    18              *DYNA INVOKE TRANSIENT TIME INTEGRATION
C    19              *OPTI FLAG THE BAND-WIDTH OPTIMIZATION
C    20              *TRAC FLAG THE DISTRIBUTED LOADING
C    21              *CREE FLAG THE CREEP STRAIN OPTION
C    22              *ANIS FLAG ANISOTROPY OPTION
C    23              *MODA MODAL ANALYSIS OPTION
C    24              *BUCK BUCKLING ANALYSIS OPTION
C    25              *THER TEMPERATURE DEPENDENT ELASTICITY OPTION
C    26              *CONS CONSTITUTIVE EQUATION SELECTION
C    27              *DIST FLAG FOR DISTRIBUTED LOAD
C    28              *DUPL DUPLICATED NODE OPTION
C    29              *REPO REPORT GENERATION INTERVAL
C    30              *TANG MODIFIED NEWTON OPTION
C    31              *UTHE USER SUBROUTINE 'UTHERM' OPTION
C    32              *SCHE TIME INTEGRATION SCHEME OPTION
C    33              *UFOR USER SUBROUTINE 'UFORCE' OPTION
C    34              *UTEM USER SUBROUTINE 'UTEMP' OPTION
C    35              *UCOE USER SUBROUTINE 'UCOEF' OPTION
C    36              *UPRE USER SUBROUTINE 'UPRESS' OPTION
C    37              *UHOO USER SUBROUTINE 'UHOOK' OPTION
C    38              *UDER USER SUBROUTINE 'UDERIV' OPTION
C    39              *UBOU USER SUBROUTINE 'UBOUN' OPTION
C    40              *PERI PERIODIC LOADING CONDITION OPTION FOR THE
C    41              *BAND BAND SOLVER ( DEFAULT)
C    42              *FRON FRONTAL SOLUTION SUBSYTEM ( OPTIONAL )
C    43              *DEFO EIGENVALUE EXTRACTION FOR THE STIFFNESS
C    44              *EMBE SUBELEMENT MESH ANALYSIS OPTION
C    45              *GMRS MULTIPLE GENERIC MODELLING REGIONS OPTION
```

```
C    46           *BEAM BEAM SECTION PARAMETER OPTION
C    47           *DISP CONVENTIONAL DISPLACEMENT METHOD
C    48           *SHIF POWER SHIFT FOR EIGEN EXTRACTION
C    49           *BFGS BFGS UPDATE FOR THE NNONLINEAR SOLUTION
C    50           *SPRI ADDED STIFFNESS, GROUND SPRING
C    51           *DASH ADDED DAMPING, DASHPOT TO GROUND
C    52           *MASS ADDED MASS
C    53           *SCEN SECANT NEWTON METHOD
C    54           *LINE LINE SEARCH
C    55           *HARM HARMONIC NODAL FORCE LOADING
C    56           *BASE HARMONIC BASE EXCITATION
C    57           *COMP COMPOSITE LAMINATE OPTION FOR ELEMENT 75
C    58           *PULS PULSE LOAD OPTION
C    59           *CONJ CONJUGATE GRADIENT ITERATION
C    60           *SHOC SHOCK SPECTRA OPTION
C    61           *SHIF POWER SPECTRAL DENSITY OPTION
C    62           *NOEC SUPRESS THE MODAL DATA ECHO PRINT
C    63           *PERT SET UP PERTURBATION SIZE FLAGS
C    64           *STIF STRESS STIFFENING OPTION
C    65           *CENT CENTRIFUGAL MASS STIFFNESS OPTION
C    66           *HARD  WORK-HARDENING OPTION FOR PLASTICITY
C    67           *FINI FINITE STRAIN OPTION
C    68           *LARG LARGE DISPLACEMENTS & ROTATIONS OPTION
C    69           *FOLL FOLLOWER FORCES OPTION
C    70           *WKSL USER SUBROUTINE 'WKSLP'
C    71           *HOUR(GLASS_CONTROL) AS THE NAME INDICATES ....
C    72           *MONI INVOKES MONITOR UTILITY
C
C *******************************************************************
C
      NOPT    =   72
      JSUBRE  =    0
      LOECHO  =    0
      IFPRNT  =    0
C
C *******************************************************************
C    SET DEFAULT VALUES
C *******************************************************************
C
      NSHIFT  =   0
      JPOST   =   0
      NSPRI   =   0
      NDASH   =   0
      NMASS   =   0
      NHARM   =   0
```

NBASE	=	0
NTIE	=	0
NDUP	=	0
JEMBED	=	0
ITHERM	=	0
NCREEP	=	1
JEMBED	=	0
JDISP	=	0
JHRGLS	=	0
JWKSLP	=	0
JISTRN	=	0
JCITER	=	0
JREPOT	=	1
ISTAT	=	1
NSBFGS	=	0
IDYNM	=	0
NGMRS	=	1
IPCONJ	=	0
JSUB	=	0
NSUB	=	0
JFRONT	=	0
JREST	=	0
JCREEP	=	0
JTEMP	=	0
NEXT	=	0
JUBOUN	=	0
NONISO	=	0
IFBFGS	=	0
IFSCNT	=	0
IFLINE	=	0
NDYNMD	=	0
IDYNMD	=	100000
NPOSMD	=	0
JTHERM	=	0
JCONST	=	2
JDYN	=	0
JEIGEN	=	0
JDEFOR	=	0
NBSECT	=	0
JFORCE	=	0
JPEROD(1)	=	0
JPEROD(2)	=	0
JUTEMP	=	0
JUCOEF	=	0
JDISTS	=	0

```
          JUHOOK    =      0
          JDERIV    =      0
          JDIST     =      0
          JOPTIM    =      0
          JPEROD(1) =      0
          JPEROD(2) =      0
          ICOMPS    =      0
          NPDPTS    =      0
          NPULSE    =      0
          NSSPTS    =      0
          NSHOCK    =      0
          NPSPTS    =      0
          NFDPTS    =      0
          JFDSXX    =      0
          JISTIF    = 999999
          JCENTM    = 999999
          JFINIT    = 999999
          JLARGE    = 999999
          JFOLOW    = 999999
C
C ***********************************************************************
C     READ TITLE CARD AND PRINT THE USUAL PROBLEM HEADER
C ***********************************************************************
C
      READ(ICREAD,1000,END=3001) TITLE
 1000 FORMAT(20A4)
     C
      CALL HEAD                                                          HOST
     1    (VERSNO,MONTH ,JDATE ,ILPRNT,ICONSL)                           HOST
C     CALL HEADER                                                        NESSUS
C    1    (VERSNO,MONTH ,JDATE ,ILPRNT,ICONSL, 2 )                       NESSUS
      CALL LINES(70,0)
      WRITE(ILPRNT,1001) TITLE
 1001 FORMAT(10X,20A4)
      NEW = .TRUE.
C
C ***********************************************************************
C     READ THE PARAMETER DATA CARDS IN THE INPUT DECK
C ***********************************************************************
C
  998 CONTINUE
C
C *** KEY-WORD INTERPRETER **********************************************
C
      CALL KEY(  NAME , NOPT , IOPT , NN , 6 , IERR  )
```

```
C
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
     &     22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,
     &     41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,
     &     60,61,62,63,64,65,66,67,68,69,70,71,72), IOPT
C
C **********************************************************************
C     OPTION 1 : *ELEM  - MAXIMUM NUMBER OF ELEMENTS IN MODEL
C **********************************************************************
C
    1 CONTINUE
                              NELEM = NN(1)
      CALL TYPEIN
     1    (IWORK ,RWORK ,IERR  ,NTYPE ,ILAST ,ILPRNT,NDIMEN)
      GO TO 998
C
C **********************************************************************
C     OPTION 2 : *NODE - MAXIMUM NUMBER OF NODES IN MODEL
C **********************************************************************
C
    2 CONTINUE
                              NNODE = NN(1)
      GO TO 998
C
C **********************************************************************
C     OPTION 3 : *BOUN - MAXIMUM NUMBER OF BOUNDARY CONDITIONS
C **********************************************************************
C
    3 CONTINUE
                              NBC   = NN(1)
      GO TO 998
C
C **********************************************************************
C     OPTION 4 : *TYIN - MAXIMUM NUMBER OF TYING CONSTRAINT
C **********************************************************************
C
    4 CONTINUE
      NTIE  = NN(1)
      NMAX  = NN(2)
      GO TO 998
C
C **********************************************************************
C     OPTION 5 : *TRAN - NUMBER OF NODAL COORDINATE TRANSFORMATIONS
C **********************************************************************
C
```

```
    5 CONTINUE
      NTRAN = NN(1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 6 : *FORC - MAXIMUM NUMBER OF NODAL FORCE ENTRIES
C
C ***********************************************************************
C
C
    6 CONTINUE
      NTRAC = NN(1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 7 : *POST - POST PROCESSING
C ***********************************************************************
C
    7 CONTINUE
      JPOST = 1
      NPOST = 1
      IF (NN(1).GT.0) NPOST = NN(1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 8 : *SUBS - INACTIVE IN VERSION 2.0
C ***********************************************************************
C
    8 CONTINUE
      JSUB = NN(1)
      NSUB = NN(2)
      IF(JSUB.NE.2) GO TO 998
      DO 108 J = 1,NSUB
      CALL FREFOR(NN,NN,3,0,0,IERR,JKEY)
      NLVSUB(NN(1)) = NN(2)
      NFRSUB(NN(1)) = NN(3)
  108 CONTINUE
      GO TO 998
C
C ***********************************************************************
C     OPTION 9 : *NEXT - EXTRNAL D.O.F. INACTIVE IN VERSION 2.0
C ***********************************************************************
C
    9 CONTINUE
      NEXT = NN(1)
```

```
      GO TO 998
C
C ***********************************************************************
C     OPTION 10 : *PRES - NODAL PRESSURE DEFINITION
C ***********************************************************************
C
   10 CONTINUE
      JPRES = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 11 : *TEMP - TEMPERATURE LOAD FLAG TO BE SET
C ***********************************************************************
C
   11 CONTINUE
      JTEMP = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 12 : *PRIN - INCREASE THE NUMBER OF PRINT OPTIONS
C ***********************************************************************
C
   12 CONTINUE
      IF (NN(1).LT.0) IFPRNT = 1
      IF (NN(1).LT.0) NPRINT = IABS(NN(1))
      IF (NN(1).GT.0) NPRINT = NN(1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 13 : *REST - RESTART TAPE FLAG TO BE SET
C ***********************************************************************
C
   13 CONTINUE
      JREST = 1
C+    JINC = NN(1)
C+    NINC = NN(2)
C
C *** EXIT IMMEDIATELY WITHOUT READING *END CARD ***********************
C
      GO TO 16
C
C ***********************************************************************
C     OPTION 14 : *LOUB - SELECTORS FOR THE NUMERICAL QUADRATURES
C ***********************************************************************
C
```

```
   14 CONTINUE
      JLOUB = 1
      JINTER = NN(1)
      JEXTRA = NN(2)
      JWEIGH = NN(3)
      JGRAM  = NN(4)
      IF(JINTER.LT.1.OR.JINTER.GT.4) JINTER = 2
      IF(JEXTRA.LT.1.OR.JEXTRA.GT.3) JEXTRA = 1
      IF(JWEIGH.LT.1.OR.JWEIGH.GT.5) JWEIGH = 1
      IF(JGRAM .LT.0.OR.JGRAM .GT.1) JGRAM  = 0
C
C --- SPECIAL TRICKS FOR INITIAL STRAIN AND CONSISTENT MASS ITERATION --
C
C     IF( NN( 6 ) .NE. 0 )  JISTRN = 1
      IF( NN( 5 ) .NE. 0 )  JCITER = NN( 5 )
      GO TO 998
C
C **********************************************************************
C     OPTION 15 : *STRE - MAXIMUM NUMBER OF STRESS BOUNDARY CONDITIONS
C **********************************************************************
C
   15 CONTINUE
      NSTRBC = NN(1)
      GO TO 998
C
C **********************************************************************
C     OPTION 17 : *TEST  FOR THE INTERNAL USE AT MARC DEVELOPMENT
C                        GROUP ONLY  - TO INVOKE THIS IS POTENTIALLY
C                        DANGEROUS
C **********************************************************************
C
   17 CONTINUE
      ISTAT  =  0
      IDYNM  =  0
      ITEST  =  1
      GO TO 998
C
C **********************************************************************
C     OPTION 18 : *DYNA - TRANSIENT TIME INTEGRATION PARAMETER
C                         COULD BE SPECIFIED IN VERSION 2.0 OR UP
C **********************************************************************
C
   18 CONTINUE
      JDYN   = NN( 1 )
      IF(JDYN .LE. 0) JDYN   = 1
```

```
      IF(JDYN .GT. 2) JDYN   = 2
      ISTAT  =  0
      IDYNM  =  1
      ITEST  =  0
      GO TO 998
C
C ***********************************************************************
C     OPTION 19 : *OPTI - BANDWIDTH OPTIMIZER ITERATION CYCLES
C ***********************************************************************
C
   19 CONTINUE
      JOPTIM = NN(1)
      IF(JOPTIM.EQ.0) JOPTIM=10
      GOTO 998
C
C ***********************************************************************
C     OPTION 20 : *TRAC - DUMMY - SAME AS OPTION 10
C ***********************************************************************
C
   20 CONTINUE
      JDIST = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 21 : *CREE - CREEP AND ITS TIME STEP CONTROL PARAMETERS
C ***********************************************************************
C
   21 CONTINUE
      JCREEP = 1
      NCREEP = 3
      ATOLER = 0.5D0
      BTOLER = 0.5D-1
      CTOLER = 0.5D-1
C
      IF( NN( 1 ) .EQ. 0 ) GO TO 2101
C
      NCREEP = NN( 1 )
C
      CALL FREFOR
     1     ( IWORK(ILAST+1) , RWORK(ILAST+1) , 0 , 3 , 0 , IERR , JKEY )
C
                J = ILAST + 1
      IF( RWORK(J) .NE. 0.0 ) CALL COPYSD ( RWORK(J) , ATOLER , 1 )
                J = ILAST + 2
      IF( RWORK(J) .NE. 0.0 ) CALL COPYSD ( RWORK(J) , BTOLER , 1 )
```

```
            J = ILAST + 3
      IF( RWORK(J) .NE. 0.0 ) CALL COPYSD ( RWORK(J) , CTOLER , 1 )
C
 2101 CONTINUE
      GO TO 998
C
C *********************************************************************
C     OPTION 21 : *ANIS - ANISOTROPIC ELASTICITY
C *********************************************************************
C
 22   CONTINUE
      NONISO=1
      GO TO 998
C
C *********************************************************************
C     OPTION 23 : *MODAL - MODAL ANALYSIS OPTION AND PARAMETER SET
C *********************************************************************
C
 23   NDYNMD=NN(1)
      NSBNC =NN(2)
      INTSTR=NN(3)
      IF ( NDYNMD .EQ. 0 )      NDYNMD =  1
      IF ( NSBNC  .EQ. 0 )      NSBNC  =  NDYNMD * 2
                                MDYNMD =  NDYNMD + 8
      IF ( NSBNC  .GT. MDYNMD) NSBNC  = MDYNMD
C
                           JEIGEN = 1
                           LDYN   = 1
                           IDYNM  = 1
                           ISTAT  = 0
C
      CALL NULINT(NN,4)
      JKEY = 0
      CALL FREFOR(NN,NN,1,0,0,IERR,JKEY)
      IF ( JKEY .EQ. 1 ) GO TO 998
      NSUPER = NN(1)
C
                           LDYN   = 2
      GO TO 998
C
C *********************************************************************
C     OPTION 24 : *BUCK - BUCKLING ANALYSIS AND PARAMETERS
C *********************************************************************
C
   24 NDYNMD = NN(1)
```

```
      NSBNC  = NN(2)
      INTSTR = NN(3)
C
      IF( NDYNMD .EQ. 0 )      NDYNMD = 1
      IF( NSBNC  .EQ. 0 )      NSBNC  = 2 * NDYNMD
                               MSBNC  = 8 + NDYNMD
                               NSBNC  = MIN0( NSBNC,MSBNC )
C
                          ISTAT  = 1
                          IDYNM  = 0
                          JEIGEN = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 25 : *THER - TEMPERATURE DEPENDENT PROPERTIES
C ***********************************************************************
C
   25 CONTINUE
      ITHERM = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 25 : *CONS - CONSTITUTIVE LAW SELECTION
C ***********************************************************************
C
   26 CONTINUE
                                      JCONST = NN(1)
      IF(JCONST.LT.0.OR.JCONST.GT.4) JCONST = 2
      GO TO 998
C
C ***********************************************************************
C     OPTION 27 : *DIST - DISTRIBUTED LOAD ( SAME AS *TRAC AND *PRES )
C ***********************************************************************
C
   27 CONTINUE
      JDIST=1
      GOTO 998
C
C ***********************************************************************
C     OPTION 28 : *DUPL - MAXIMUM NUMBER OF DUPLICATED NODES
C ***********************************************************************
C
   28 CONTINUE
      NDUP=NN(1)
      GOTO 998
```

```
C
C ***********************************************************************
C     OPTION 29 : *REPO - REPORT GENERATION INTERVAL TO BE SET
C ***********************************************************************
C
   29 CONTINUE
      JREPOT = NN( 1 )
      IF(NN( 1 ) .EQ. 0) JREPOT=1
      GO TO 998
C
C ***********************************************************************
C     OPTION 30 : *TANG - MODIFIED NEWTON METHOD WITH TANGENT MATRIX
C                         SPECIFICATION
C ***********************************************************************
C
   30 CONTINUE
      JTANGE = NN( 1 )
      GO TO 998
   31 CONTINUE
      JTHERM = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 32 : *SCHE - TIME STEPPING SCHEME PARAMETER OPTION
C ***********************************************************************
C
   32 CONTINUE
      DALPHA = 0.5D0
      DBETA  = 0.25D0
      DGAMMA = 0.5D0
      CALL FREFOR(IWORK(ILAST+1),RWORK(ILAST+1),0,3,0,IERR,JKEY)
      CALL COPYSD (RWORK(ILAST+1),DALPHA,1)
      CALL COPYSD (RWORK(ILAST+2),DBETA ,1)
      CALL COPYSD (RWORK(ILAST+3),DGAMMA,1)
      GO TO 998
C
C ***********************************************************************
C     OPTIONS 33, 34, 35, 36, 37, 38, 39 - FLAGS FOR USER SUBROUTINES
C ***********************************************************************
C
   33 CONTINUE
      JFORCE = 1
      GO TO 998
C
   34 CONTINUE
```

```
      JUTEMP = 1
      GO TO 998
C
   35 CONTINUE
      JUCOEF = 1
      GO TO 998
C
   36 CONTINUE
      JDISTS = 1
      GO TO 998
C
   37 CONTINUE
      JUHOOK = 1
      GO TO 998
C
   38 CONTINUE
      JDERIV = 1
      GO TO 998
C
   39 CONTINUE
      JUBOUN = 1
      GO TO 998
C
C **********************************************************************
C     OPTION 40 : *PERI - PERIODIC LOADING APPLICABLE ONLY FOR TRANSIENT
C                         DYNAMICS
C **********************************************************************
C
   40 CONTINUE
      JPEROD( 1 ) = NN( 1 )
      JPEROD( 2 ) = NN( 2 )
      GO TO 998
C
C **********************************************************************
C     OPTION 41 : *BAND - BAND MATRIX SOLVER ( DEFAULT )
C **********************************************************************
C
   41 CONTINUE
                JBAND  = 1
                JFRONT = 0
      GO TO 998
C
C **********************************************************************
C     OPTION 42 : *FRON - FRNTAL SOLUTION
C **********************************************************************
```

```
C
   42 CONTINUE
                 JBAND  = 0
                 JFRONT = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 43 : *DEFO - EIGENVALUE EXTRACTION FOR THE STIFFNESS
C                         MATRIX FLAGGED
C ***********************************************************************
C
   43 CONTINUE
                JDEFOR = 1
                IDYNMD = NN( 1 )
                NDYNMD = NN( 2 )
                NSBNC  = NN( 3 )
                INTSTR = NN( 4 )
C
      IF ( NDYNMD .EQ. 0 ) NDYNMD = 1
                           NSBNC  = 2 * NDYNMD
C
                           MSBNC  = 8 + NDYNMD
      IF ( NDYNMD .GT. 8 ) NSBNC  = MSBNC
C
                          ISTAT  = 1
                          IDYNM  = 0
                          JEIGEN = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 44 : *EMBE - EMBEDDED MULTIPLE SINGULARITIES
C ***********************************************************************
C
   44 CONTINUE
                              JEMBED   =   1
      IF( NN( 1 ) .NE. 0 )    JSUBRE   =   1
C
      GO TO 998
C
C ***********************************************************************
C     OPTION 45 : *GMRS - MAXIMUM NUMBER OF MULTIPLE GENERIC REGIONS
C ***********************************************************************
C
   45 CONTINUE
                         NGMRS = NN( 1 )
```

```
      IF( NGMRS .EQ. 0 ) NGMRS =   1
      GO TO 998
C
C ************************************************************************
C     OPTION 46 : *BEAM - MAXIMUM NUMBER OF BEAM SECTION DATA
C ************************************************************************
C
   46 CONTINUE
      NBSECT = NNODE
      GO TO 998
C
C ************************************************************************
C     OPTION 47 : *DISP - DISPLACEMENT METHOD OPTION
C ************************************************************************
C
   47 CONTINUE
               JDISP  = 1
      GO TO 998
C
C ************************************************************************
C     OPTION 48 : *POWE - POWER SHIFT FOR EIGEN ANALYSIS
C ************************************************************************
C
   48 CONTINUE
                          NSHIFT = NN( 1 )
      IF( NSHIFT .EQ. 0 ) NSHIFT =    1
      GO TO 998
C
C ************************************************************************
C     OPTION 49 : *BFGS - BFGS UPDATE FOR QUASI-STATIC NONLINEAR
C                         ANALYSIS
C ************************************************************************
C
   49 CONTINUE
                          IFBFGS =   1
                          NSBFGS =  10
      IF( NN(1) .NE. 0 )  NSBFGS = NN( 1 )
      GO TO 998
C
C ************************************************************************
C     OPTION 50 : *SPRI - ADDED STIFFNESS, GROUND SPRING
C ************************************************************************
C
   50 CONTINUE
                          NSPRI  = NN(1)
```

```
      IF ( NSPRI .EQ. 0 ) NSPRI  =    1
      GO TO 998
C
C **********************************************************************
C     OPTION 51 : *DASH - ADDED DAMPING, DASHPOT TO GROUND
C **********************************************************************
C
   51 CONTINUE
                            NDASH  = NN(1)
      IF ( NDASH .EQ. 0 ) NDASH  =    1
      GO TO 998
C
C **********************************************************************
C     OPTION 52 : *MASS - ADDED MASS
C **********************************************************************
C
   52 CONTINUE
                            NMASS  = NN(1)
      IF ( NMASS .EQ. 0 ) NMASS  =    1
      GO TO 998
C
C ***********************************************************************
C     OPTION 53 : *SECA - SECANT NEWTON METHOD OPTION
C ***********************************************************************
C
   53 CONTINUE
                          IFSCNT   =   1
      GO TO 998
C
C ***********************************************************************
C     OPTION 54 : *LINE - LINE SEARCH OPTION
C ***********************************************************************
C
   54 CONTINUE
                          IFLINE   =   1
      GO TO 998
C
C ***********************************************************************
C     OPTION 55 : *HARM - HARMONIC NODAL FORCE LOADING
C ***********************************************************************
C
   55 CONTINUE
      NHARM = NN(1)
      IF (NHARM .EQ. 0) NHARM = 1
      CALL FREFOR(IWORK(ILAST+1),RWORK(ILAST+1),0,1,0,IERR,JKEY)
```

```
      CALL COPYSD(RWORK(ILAST+1),OMEGH,1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 56 : *BASE - HARMONIC BASE EXCITATION
C ***********************************************************************
C
   56 CONTINUE
      NBASE = NN(1)
      IF (NBASE .EQ. 0) NBASE = 1
      CALL FREFOR(IWORK(ILAST+1),RWORK(ILAST+1),0,1,0,IERR,JKEY)
      CALL COPYSD(RWORK(ILAST+1),OMEGB,1)
      GO TO 998
C
C ***********************************************************************
C     OPTION 57 : COMPOSITE LAMINATE OPTION
C ***********************************************************************
C
   57 CONTINUE
                                 ICOMPS  =   1
      CALL COMPDF
     1     (IWORK ,RWORK ,IERR   ,NTYPE ,ILAST ,ILPRNT)
C
      GO TO 998
C
C
C ***********************************************************************
C     OPTION 58 : *PULS - PULSE LOAD OPTION
C ***********************************************************************
C
   58 CONTINUE
      NPDPTS = NN(1)
      NPULSE = NN(2)
      IF (NPDPTS .LT. 2) NPDPTS = 2
      IF (NPULSE .EQ. 0) NPULSE = 1
      GO TO 998
C
C ***********************************************************************
C     OPTION 59 : PRECONDITIONED CONJUGATE GRADIENT ITERATION OPTION
C ***********************************************************************
C
   59 CONTINUE
                  IPCONJ     =      1
                  IFLINE     =      1
      GO TO 998
```

```
C
C *******************************************************************
C     OPTION 60 : *SHOC - SHOCK SPECTRA OPTION  (INACTIVE)
C *******************************************************************
C
   60 CONTINUE
      CALL LINES(1,1)
C     WRITE(ILPRNT,6000) 61,'SHOC'
C     WRITE(ICONSL,6000) 61,'SHOC'
      IERR = IERR+1
      NSSPTS = NN(1)
      IF ( NSSPTS .EQ. 0 ) NSSPTS = 10
      LDYN = 3
      GO TO 998
C
C *******************************************************************
C     OPTION 61 : *POWE - POWER SPECTRAL DENSITY OPTION
C *******************************************************************
C
   61 CONTINUE
      NPSPTS = NN(1)
      NFDPTS = NN(2)
      JFDSXX = NN(3)
      IF ( NPSPTS .EQ. 0 ) NPSPTS = 10
      IF ( NFDPTS .EQ. 0 ) NFDPTS = 10
      LDYN = 4
      GO TO 998
C
C -------------------------------------------------------------------
C
   62 CONTINUE
                LOECHO = 1
      GO TO 998
C
C *******************************************************************
C     OPTION 63 : *PERT - SET UP PERTURBATION FLAGS
C *******************************************************************
C
   63 CONTINUE
      CALL LINES(1,1)                                              HOST
      WRITE(ILPRNT,6000) 63,'PERT'                                 HOST
      WRITE(ICONSL,6000) 63,'PERT'                                 HOST
      IERR = IERR+1                                                HOST
C     CALL PERSIZ( NN(1), IERR )                                   NESSUS
      GO TO 998
```

```
C
C ************************************************************************
C     OPTION 64 : *STIF - STRESS STIFFENING OPTION
C ************************************************************************
C
   64 CONTINUE
      JISTIF = NN(1)
      IF( NN(1).EQ.0 ) JISTIF = 1
      GO TO 998
C
C ************************************************************************
C     OPTION 65 : *CENT - CENTRIFUGAL MASS STIFFNESS OPTION
C ************************************************************************
C
   65 CONTINUE
      JCENTM = NN(1)
      GO TO 998
C
C *************************************************************************
C     OPTION 66 : *HARD - WORK-HARDENING OPTION FOR PLASTICITY
C *************************************************************************
C
   66 CONTINUE
      NHARD = NN(1)
      IF (NN(1) .EQ. 0) NHARD = 1
      GO TO 998
C
C *************************************************************************
C     OPTION 67 : *FINIT - FINITE STRAIN OPTION
C **************************************************************************
C
   67 CONTINUE
      JFINIT = NN(1)
      GO TO 998
C
C **************************************************************************
C     OPTION 68 : *LARG - LARGE DISPLACEMENTS AND ROTATIONS OPTION
C **************************************************************************
C
   68 CONTINUE
      JLARGE = NN(1)
      GO TO 998
C
C **************************************************************************
C     OPTION 69 : *FOLL - FOLLOWER FORCE OPTION
```

```
C ***********************************************************************
C
   69 CONTINUE
      JFOLOW = NN(1)
      GO TO 998
C
C **********************************************************************
C    OPTION 70 : *WLSKP : FLAGS THE USER SUBROUTINE FOR WORKHARDENING
C **********************************************************************
C
   70 CONTINUE
      JWKSLP = 1
      GO TO 998
C
C **********************************************************************
C     OPTION 71 : *HOUR ... HOURGLASS CONTROL FLAG IN A SPECIAL WAY
C **********************************************************************
C
   71 CONTINUE
      JHRGLS  =  1
      GO TO 998
C
C **********************************************************************
C     OPTION 72 : *MONI - TURN ON THE MONITOR UTILITY
C **********************************************************************
C
   72 CONTINUE
      NMONIT = NN(1)
      IF (NMONIT .LT. 1) NMONIT = 1
      GO TO 998
C
C *** NORMAL EXIT BY READING '*END' CARD *******************************
C
   16 CONTINUE
                 IF( LOECHO .NE. 0 ) NOECHO = 1
C
C *** A FEW POSSIBLE CONTRADICTIONS IN PARAMETER DATA ARE CHECKED HERE
C
      IF( IFBFGS .EQ. 1 .AND. IFSCNT .EQ. 1 ) CALL QUIT
     1    ('CONT','RADI','CTIO','N DE','TECT','ED  ', 1 )
C
      IF( IFBFGS .EQ. 1 .AND. IPCONJ .EQ. 1 ) CALL QUIT
     1    ('CONT','RADI','CTIO','N DE','TECT','ED  ', 1 )
C
      IF( IPCONJ .EQ. 1 .AND. IFSCNT .EQ. 1 ) CALL QUIT
```

```
     1    ('CONT','RADI','CTIO','N DE','TECT','ED  ', 1 )
C
      IF( JFINIT.LT.JLARGE                     ) CALL QUIT
     &    ('*FIN','I ST','ARTS',' B4 ','*LAR','GE  ', 0 )
C
      IF( JLARGE.EQ.999999 )                     T H E N
      IF( JFOLOW.NE.999999                     ) CALL QUIT
     &    ('*FOL','L BU','T NO','*LAR','G   ','    ', 0 )
      IF( JFINIT.NE.999999 )                     CALL QUIT
     &    ('*FIN','I BU','T NO','*LAR','G   ','    ', 0 )
                                                 E N D I F
C
      IF( JLARGE.NE.999999 .AND. JISTIF.EQ.999999 ) CALL PRWARN
     &  ( 'LARGE DISPL OPTION WITHOUT INITIAL STRESS OPTION')
C
      R E T U R N
C
C *** CONTROL DATA ERROR ***********************************************
C
 3001 CONTINUE
      WRITE(ICONSL,9000)
      CALL QUIT('INPU','T   ','    ','    ','    ','    ',1)
C
C *** FORMAT STATEMENTS ************************************************
C
 6000 FORMAT(/,1X,23H***ERROR***     OPTION ,I2,3H  *,A4,11H NOT ACTIVE)
 9000 FORMAT(//,28H ********** ERROR **********,/,14H NO INPUT FILE,//)
C
      S T O P
      END
```

An example of a typical bulk data reader is the subprogram to read and store the nodal coordinates:

```
C=SUBROUTINE=COORIN  CALLED BY SUBROUTINE 'DATIN2'
      SUBROUTINE COORIN
     1    (INT   ,REA   ,IERR  ,NCRD  ,ILAST ,ILPRNT,NNODE,INOD  ,
     2     MAXCRD,ICHECK)
C
C ********************************************************************
C
C     READ IN COORDINATE DATA
C
C     IERR    ERROR FLAG
C     NCRD    NUMBER OF COORDINATES PER NODE
```

```
C     ILAST   LAST WORD FOR INPUT
C     KW      OUTPUT DEVICE
C     NNODE   NUMBER OF NODES IN MESH
C     INOD    POINTER TO COORDINATES
C     MAXCRD  MAXIMUM NUMBER OF COORDINATES PER NODE
C     ICHECK  ENABLE/DISABLE NODAL THICKNESS CHECK
C
C ***********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
               REAL*4 REA
C
C ***********************************************************************
C
      COMMON / MACHIN / IDP
C
C ***********************************************************************
C
      DIMENSION INT( 1 ) , REA( 1 )
C
C ***********************************************************************
C
      IS1  = INOD - MAXCRD*IDP
      JKEY = -1
C
C *** CHECK IF THE CORE IS ALLOCATED FOR THE COORDINATE DATA ***********
C
      IF ( NNODE .GT. 0 ) GO TO  208
C
                                          CALL LINES( 1 , 1 )
      WRITE(ILPRNT,9108)
 9108 FORMAT(2X,48H***ERROR***     NO SPACE RESERVED FOR COORDINATE,
     1        5H DATA)
                                            IERR = IERR + 1
C
C *** CHECK THE NUMBER OF DATA ENTRIES **********************************
C
  208 CONTINUE
      IF ( NCRD .EQ. 0 ) NCRD = MAXCRD
C
C *** MCRD: COUNTER OF THE NUMBER OF ACTUAL COORDINATE DATA ENTRIES ****
C
                            MCRD = NCRD
      IF ( MAXCRD .EQ. 7 ) MCRD = 3
C
```

```
      IF ( NCRD .LE. MAXCRD ) GO TO  607
                                            CALL LINES( 1 , 1 )
      WRITE(ILPRNT,9208) NCRD,MAXCRD
 9208 FORMAT(2X,47H***ERROR***     NUMBER OF COORDINATE DIRECTIONS,I5,
     *13H GREATER THAN,I5)
                                            IERR = IERR + 1
C
C *** READ NUMERIC DATA LINE *******************************************
C
  607                                       ICOUNT = 0
C
                                            TNEW   = 0.D0
                                            TOLD   = 0.D0
  608 CONTINUE
                                            ICOUNT = ICOUNT + 1
C
      CALL NULINT(INT(ILAST+1),MCRD+1)
      CALL FREFOR(INT(ILAST+1),REA(ILAST+1),1,NCRD,0,IERR,JKEY)
C
C *** IF THE CURRENT LINE IS THE KEYWORD DATA THEN RETURN **************
C
      IF ( JKEY .EQ. 1 ) GO TO  108
C
C **********************************************************************
C     PROCESS THE NODAL COORDINATE DEFINITION DATA LINE
C **********************************************************************
C
      TOLD = TNEW
      K    = INT(ILAST+1)
      TNEW = REA(ILAST+5)
C
C *** IF THICKNESS IS NOT GIVEN THE LAST NONZERO VALUE IS TAKEN ********
C
      IF ( TNEW .EQ. 0.D0 ) TNEW = TOLD
C
C *** RANGE CHECK ******************************************************
C
      IF ( K .GT. 0 .AND. K .LE. NNODE ) GO TO  308
C
C *** ERROR MESSAGES ***************************************************
C
                                            CALL LINES( 1 , 1 )
      IF ( K .LE. 0 ) WRITE(ILPRNT,9601) K
 9601 FORMAT(2X,27H***ERROR***     NODE NUMBER,I5,13H NON-POSITIVE)
C
```

```
C ***
      IF ( K .GT. NNODE ) WRITE(ILPRNT,9701) K, NNODE
 9701 FORMAT(2X,27H***ERROR***     NODE NUMBER,I5,13H GREATER THAN,I5)
C
                                                IERR = IERR + 1
  308 CONTINUE
C
C *** THICKNESS DATA CHECK FOR SHELL ELEMENTS ***************************
C
      IF ( MAXCRD .NE. 7 .OR. ICHECK .EQ. 0   ) GO TO 8000
      IF ( ICOUNT .NE. 1 .OR. TNEW .NE. 0.D0 ) GO TO 8000
C
C *** ERROR MESSAGE FOR THE THICKNESS NOT DEFINED ********************
C
                                              CALL LINES( 1 , 1 )
      WRITE(ILPRNT,9753)
 9753 FORMAT(2X,'***ERROR*** SHELL THICKNESS NOT SPECIFIED IN THE ',
     1          'FIRST DATA LINE')
                                                IERR = IERR + 1
C
C *** ADRESSES FOR THE CURRENT (K-TH) NODAL COORDINATES ****************
C
 8000                                                CONTINUE
      IS2 = IS1 + K*MAXCRD*IDP
      IS3 = IS2 + 6       *IDP
C
C *** STORE THE COORDINATE DATA *****************************************
C
      CALL COPYSD ( REA(ILAST+2) , REA(IS2) , MCRD )
C
      IF ( MAXCRD .NE. 7 ) GO TO 408
C
C *** STORE THE SHELL THICKNESS DATA AT NODE ***************************
C
      CALL COPY   ( TNEW         , REA(IS3) ,    1   )
C
  408 CONTINUE
  508 CONTINUE
C
C *** BACK TO THE FREE FORMAT READER AND PROCESS THE NEXT LINE *********
C
      GO TO  608
C
C *** EXIT **************************************************************
C
```

```
  108 CONTINUE
      RETURN
      END
```

3.3 Restart File

The restart file is designed to store all the information necessary to resume execution of an incremental analysis or to use the final result stored in the file for starting a new set of calculations. The alteration of certain parameter data is supported by the following SUBROUTINES: SAVER to write and RESTRT to read the restart file:

```
C ... SUBROUTINE SAVER ... RESTART FILE WRITER
C
      SUBROUTINE SAVER( RWORK, IWORK, ISIZE )
C
C *********************************************************************
C **                                                                 **
C **  WRITES A BINARY FILE FOR LATER RESTART                         **
C **                                                                 **
C *********************************************************************
C
C     IWORK :  INTEGER WORKSPACE
C     RWORK :  REAL WORKSPACE
C     ISIZE :  SIZE OF WORKSPACE
C
C *********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
               REAL*4  RWORK
C
      DIMENSION IWORK(ISIZE), RWORK(ISIZE)
C
      COMMON / ADDVAL / ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / AUTOIN / CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
     1                  JADAP ,NCREEP,SCALE
      COMMON / BODYFR / POINTS(     3,      2)
      COMMON / BSECT  / IBSECT,KBSECT
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
```

```
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / COMPND / NXSTAT,NXSOLV,NXINTG,NXMODL,NXBCKL,NXSUPR,
     1                  NXREQN,NXDUM1,NXDUM2,NXDUM3,NXDUM4
      COMMON / COUNT  / LININC,LINTOT,NOECHO
      COMMON / DAMP   / DAMPF(3)
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / MODSUP / IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      COMMON / HARMON / OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
     1                  ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                  ICCMAT
      COMMON / INCCON / FACTOR,INCFLG(21)
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / LOUBIN / JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
     1                  JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / PERPAR / IPTYPE(    32),NPTYPE,NPVARS,NPSETS,JPERT ,   NESSUS
     1                  NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012     NESSUS
      COMMON / PERPTR / IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,    NESSUS
     1                  IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,    NESSUS
     2                  IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK     NESSUS
      COMMON / PERDAT / IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,    NESSUS
     1                  IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,    NESSUS
     2                  IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND     NESSUS
      COMMON / PERIOD / JPEROD(    2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / POSTPN / IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(  6   )
```

```
      COMMON / POWER  / IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                  IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                  IPSF  ,IPSD
      COMMON / PREPAR / NRFPTS,NRFSDS,NRFLIN,NRFINT,NRFREA,NRFSEL,     NESSUS
     1                  NRFCMP,NRFDOF,NRFSIZ,LRFEND,JRFTYP,JCOUNT,     NESSUS
     2                  JCFLAG,JRFSEL,JRFDOM,NVKEPT,MORE               NESSUS
      COMMON / PREPTR / IRFPTS,IRFINT,IRFMNR,IRFSDR,IRFSEL,IRFSET,     NESSUS
     1                  IRFMNV,IRFSDV,IRFPTV,IRFVEC,IRFVAL,IRFNRM,     NESSUS
     2                  IRFCOR,IRFID ,IRFWRK                           NESSUS
      COMMON / PULSES / IPULSE,KPULSE,IPDTIM,IPDFOR
      COMMON / RESULT / MANVAR(    7 ),JPR   ,ICOM1 ,NCOMP
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                  IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                  ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                  IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                  IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                  IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                  IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                  IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                  IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                  IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                  IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                  IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                  IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                  IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                  IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                  KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                  KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                  IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                  KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                  IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                  IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                  IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                  KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                  KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      COMMON / SUBELM / ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
```

```
      COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
     1                  NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
      COMMON / SUBSTR / NLVSUB(    10),NFRSUB(    10)
      COMMON / SHIFT  / ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
      COMMON / TIME   / TIMINC,TOTINC,RUNTIM
      COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
      COMMON / TOLER  / RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
     1                  ENGTOR,ENGNRM
      COMMON / VRTDSK / ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
     1                  IVPAGE,NVPAGE,IVSTRT,IKRO  ,ILCOL ,IPIVCO,
     2                  IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND ,
     3                  ISRECD,IERECD
      COMMON / ZPRINT / MAXCOL,NELCOL,LAYPRT,JTENSR
C
C ***********************************************************************
C **  REWIND THE RESTART FILE TO OVERWRITE WITH THE NEWEST DATA       **
C ***********************************************************************
C
      REWIND(IRSTRT)
C
C ***********************************************************************
C **  FIRST STORE THE CONTENTS OF THE COMMON BLOCKS                   **
C ***********************************************************************
C
      WRITE (IRSTRT)    ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      WRITE (IRSTRT)    ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      WRITE (IRSTRT)    CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
     1                  JADAP ,NCREEP,SCALE
      WRITE (IRSTRT)    POINTS(     3,     2)
      WRITE (IRSTRT)    IBSECT,KBSECT
      WRITE (IRSTRT)    JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      WRITE (IRSTRT)    NXSTAT,NXSOLV,NXINTG,NXMODL,NXBCKL,NXSUPR,
     1                  NXREQN,NXDUM1,NXDUM2,NXDUM3,NXDUM4
```

```
      WRITE (IRSTRT)   LININC,LINTOT,NOECHO
      WRITE (IRSTRT)   DAMPF(3)
      WRITE (IRSTRT)   IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                 IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      WRITE (IRSTRT)   IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      WRITE (IRSTRT)   IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                 JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      WRITE (IRSTRT)   OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
     1                 ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                 ICCMAT
      WRITE (IRSTRT)   FACTOR,INCFLG(20)
      WRITE (IRSTRT)   MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                 MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                 MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                 MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                 MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      WRITE (IRSTRT)   JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
     1                 JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      WRITE (IRSTRT)   NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                 NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                 NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                 NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                 NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                 NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                 NSUMCH,NPAR38,NMONIT,NPAR40,NPAR41,NPAR42,
     7                 NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      WRITE (IRSTRT)   IPTYPE(    32),NPTYPE,NPVARS,NPSETS,JPERT ,       NESSUS
     1                 NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012         NESSUS
      WRITE (IRSTRT)   IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,        NESSUS
     1                 IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,        NESSUS
     2                 IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK         NESSUS
      WRITE (IRSTRT)   IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,        NESSUS
     1                 IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,        NESSUS
     2                 IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND         NESSUS
      WRITE (IRSTRT)   JPEROD(   2 ),IPDISP,IPFORC,INDISP,INFORC
      WRITE (IRSTRT)   IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(  6   )
      WRITE (IRSTRT)   IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                 IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                 IPSF  ,IPSD
      WRITE (IRSTRT)   NRFPTS,NRFSDS,NRFLIN,NRFINT,NRFREA,NRFSEL,        NESSUS
     1                 NRFCMP,NRFDOF,NRFSIZ,LRFEND,JRFTYP,JCOUNT,        NESSUS
     2                 JCFLAG,JRFSEL,JRFDOM,NVKEPT,MORE                  NESSUS
      WRITE (IRSTRT)   IRFPTS,IRFINT,IRFMNR,IRFSDR,IRFSEL,IRFSET,        NESSUS
     1                 IRFMNV,IRFSDV,IRFPTV,IRFVEC,IRFVAL,IRFNRM,        NESSUS
     2                 IRFCOR,IRFID ,IRFWRK                              NESSUS
```

```
 WRITE (IRSTRT)     IPULSE,KPULSE,IPDTIM,IPDFOR
 WRITE (IRSTRT)     MANVAR(   7 ),JPR   ,ICOM1 ,NCOMP
 WRITE (IRSTRT)     IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
1                   ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
2                   IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
 WRITE (IRSTRT)     INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
 WRITE (IRSTRT)     IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
1                   ISBC  ,ISBCR
 WRITE (IRSTRT)     IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
1                   IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
2                   IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
3                   IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
4                   IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
5                   IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
6                   IEQCSI,IPREF ,IDSX3 ,IYIELD,IST441,IST442,
7                   IST443,IST444,IST445,IST446,IST447,IST448
 WRITE (IRSTRT)     IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
1                   IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
2                   IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
3                   IMASDI,IMASUP,IST521,IST522,IST523,IST524
 WRITE (IRSTRT)     IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
1                   IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
2                   IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
3                   KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
4                   KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
5                   IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
6                   KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
7                   IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
8                   IST649,IST650,IST651,IST652,IST653,IST654
 WRITE (IRSTRT)     ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
 WRITE (IRSTRT)     KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
1                   IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
2                   KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
 WRITE (IRSTRT)     KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
1                   KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
 WRITE (IRSTRT)     ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
 WRITE (IRSTRT)     NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
1                   NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
 WRITE (IRSTRT)     NLVSUB(   10),NFRSUB(   10)
 WRITE (IRSTRT)     ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
 WRITE (IRSTRT)     TIMINC,TOTINC,RUNTIM
 WRITE (IRSTRT)     DALPHA,DBETA ,DGAMMA
 WRITE (IRSTRT)     RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
1                   ENGTOR,ENGNRM
 WRITE (IRSTRT)     ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
```

```
     1                  IVPAGE,NVPAGE,IVSTRT,IKRO   ,ILCOL ,IPIVCO,
     2                  IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND ,
     3                  ISRECD,IERECD
      WRITE (IRSTRT)    MAXCOL,NELCOL,LAYPRT,JTENSR
C
C *********************************************************************
C **  NEXT STORE THE DATA IN BLANK COMMON IN 64 WORD BLOCKS          **
C *********************************************************************
C
      NWORDS = 64
      NRECDS = ILAST / NWORDS
C
      DO 600 JREC = 1, NRECDS
        IBEG = 1+(JREC-1)*NWORDS
        IEND = JREC*NWORDS
        WRITE (IRSTRT)  ( IWORK(JJ), JJ=IBEG,IEND )
  600 CONTINUE
C
      IBEG = 1+NRECDS*NWORDS
      IEND = ILAST
      IF (IBEG .NE. IEND)
     1  WRITE (IRSTRT)  ( IWORK(JJ), JJ=IBEG,IEND )
C
C *********************************************************************
C **  PRINT MESSAGE AND GO ON TO THE NEXT INCREMENT                   **
C *********************************************************************
C
      CALL TIMOUT ('REST','ART ','FILE',' GEN','ERAT','ED  ')
      RETURN
      END

C ... SUBROUTINE RESTRT ... RESTART FILE READER
C
      SUBROUTINE RESTRT( RWORK, IWORK, ISIZE )
C
C *********************************************************************
C **                                                                 **
C **  READS-IN A BINARY FILE FOR PROBLEM RESTART                     **
C **                                                                 **
C *********************************************************************
C
C     IWORK :  INTEGER WORKSPACE
C     RWORK :  REAL WORKSPACE
C     ISIZE :  SIZE OF WORKSPACE
C
```

```
C ********************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
               REAL*4  RWORK
C
      DIMENSION IWORK(ISIZE), RWORK(ISIZE)
C
      COMMON / ADDVAL / ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / AUTOIN / CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
     1                  JADAP ,NCREEP,SCALE
      COMMON / BODYFR / POINTS(     3,      2)
      COMMON / BSECT  / IBSECT,KBSECT
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / COMPND / NXSTAT,NXSOLV,NXINTG,NXMODL,NXBCKL,NXSUPR,
     1                  NXREQN,NXDUM1,NXDUM2,NXDUM3,NXDUM4
      COMMON / COUNT  / LININC,LINTOT,NOECHO
      COMMON / DAMP   / DAMPF(3)
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / MODSUP / IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      COMMON / HARMON / OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
     1                  ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                  ICCMAT
      COMMON / INCCON / FACTOR,INCFLG(21)
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / LOUBIN / JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
```

```
     1                   JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                   NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                   NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                   NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                   NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                   NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                   NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                   NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / PERPAR / IPTYPE(   32),NPTYPE,NPVARS,NPSETS,JPERT ,      NESSUS
     1                   NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012      NESSUS
      COMMON / PERPTR / IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,      NESSUS
     1                   IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,     NESSUS
     2                   IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK      NESSUS
      COMMON / PERDAT / IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,      NESSUS
     1                   IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,     NESSUS
     2                   IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND      NESSUS
      COMMON / PERIOD / JPEROD(   2 ),IPDISP,IPFORC,INDISP,INFORC
      COMMON / POSTPN / IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(  6  )
      COMMON / POWER  / IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                   IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                   IPSF  ,IPSD
      COMMON / PREPAR / NRFPTS,NRFSDS,NRFLIN,NRFINT,NRFREA,NRFSEL,      NESSUS
     1                   NRFCMP,NRFDOF,NRFSIZ,LRFEND,JRFTYP,JCOUNT,     NESSUS
     2                   JCFLAG,JRFSEL,JRFDOM,NVKEPT,MORE               NESSUS
      COMMON / PREPTR / IRFPTS,IRFINT,IRFMNR,IRFSDR,IRFSEL,IRFSET,      NESSUS
     1                   IRFMNV,IRFSDV,IRFPTV,IRFVEC,IRFVAL,IRFNRM,     NESSUS
     2                   IRFCOR,IRFID ,IRFWRK                           NESSUS
      COMMON / PULSES / IPULSE,KPULSE,IPDTIM,IPDFOR
      COMMON / RESULT / MANVAR(   7 ),JPR   ,ICOM1 ,NCOMP
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                   ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                   IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                   ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                   IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                   IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                   IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                   IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                   IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                   IEQCSI,IPREF ,IDSX3 ,IYIELD,IDFINC,IDFTOT,
     7                   IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
```

```
     1                   IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                   IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                   IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                   IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                   IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                   KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                   KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                   IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                   KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                   IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                   IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                   IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                   KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / START9 / KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                   KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      COMMON / SUBELM / ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
      COMMON / SUBTYP / NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
     1                   NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
      COMMON / SUBSTR / NLVSUB(    10),NFRSUB(    10)
      COMMON / SHIFT  / ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
      COMMON / TIME   / TIMINC,TOTINC,RUNTIM
      COMMON / TMARCH / DALPHA,DBETA ,DGAMMA
      COMMON / TOLER  / RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
     1                   ENGTOR,ENGNRM
      COMMON / VRTDSK / ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
     1                   IVPAGE,NVPAGE,IVSTRT,IKRO  ,ILCOL ,IPIVCO,
     2                   IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND ,
     3                   ISRECD,IERECD
      COMMON / ZPRINT / MAXCOL,NELCOL,LAYPRT,JTENSR
C
C ***********************************************************************
C **  READ BACK THE CONTENTS OF THE COMMON BLOCKS                      **
C ***********************************************************************
C
      READ  (IRSTRT)    ISPRI ,KSPRI ,IDASH ,KDASH ,IMASS ,KMASS
      READ  (IRSTRT)    ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                   IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                   PI    ,LINE  ,LINE2
      READ  (IRSTRT)    CURPER,TOTPER,ARCLEN,ATOLER,BTOLER,CTOLER,
     1                   JADAP ,NCREEP,SCALE
      READ  (IRSTRT)    POINTS(     3,     2)
      READ  (IRSTRT)    IBSECT,KBSECT
```

```
      READ   (IRSTRT)   JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
     6                  JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                  IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                  IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                  JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                  JWKSLP,JPRES ,JCDUM2,JCDUM3
      READ   (IRSTRT)   NXSTAT,NXSOLV,NXINTG,NXMODL,NXBCKL,NXSUPR,
     1                  NXREQN,NXDUM1,NXDUM2,NXDUM3,NXDUM4
      READ   (IRSTRT)   LININC,LINTOT,NOECHO
      READ   (IRSTRT)   DAMPF(3)
      READ   (IRSTRT)   IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      READ   (IRSTRT)   IMFOR0,IMDIS0,IMVEL0,IMFOR1,IMDIS1,IMVEL1
      READ   (IRSTRT)   IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      READ   (IRSTRT)   OMEGH ,IHARM ,KHARM ,OMEGB ,IBASE ,KBASE ,
     1                  ICNFOR,ICMFOR,ICMRES,ICHHFN,ICBHFN,ICBEXC,
     2                  ICCMAT
      READ   (IRSTRT)   FACTOR,INCFLG(20)
      READ   (IRSTRT)   MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      READ   (IRSTRT)   JLOUB ,JINTER,JEXTRA,JWEIGH,JSUBRE,JISTRN,
     1                  JCITER,JHRGLS,JGRAM ,LOUB03,LOUB04,LOUB05
      READ   (IRSTRT)   NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NPAR38,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      READ   (IRSTRT)   IPTYPE(    32),NPTYPE,NPVARS,NPSETS,JPERT ,        NESSUS
     1                  NPVCON,NPP008,NPP009,NPP010,NPP011,NPP012          NESSUS
      READ   (IRSTRT)   IMEANS,ISTDEV,IPDATA,IVTYPE,ISKIP ,IREDEF,         NESSUS
     1                  IDINC0,IREAC0,IRESD0,IDGRP ,ISTIF0,IMASS0,         NESSUS
     2                  IPP013,IOMEG0,IOMEGP,IOMEGK,IETAK ,IZETAK          NESSUS
      READ   (IRSTRT)   IXCOOR,IXCHAR,IXFORC,KXFORC,IXDIST,KXDIST,         NESSUS
```

```
     1                   IXTEMP,JXTEMP,IXBEAM,IXFVEC,IXSPRI,KXSPRI,     NESSUS
     2                   IXPRES,IXPREF,IXP015,IXP016,IPWBEG,IPWEND      NESSUS
      READ  (IRSTRT)     JPEROD(   2 ),IPDISP,IPFORC,INDISP,INFORC
      READ  (IRSTRT)     IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(  6  )
      READ  (IRSTRT)     IELPHI,IELTNM,IEPSMO,ISIGMO,IHFN  ,IHFC  ,
     1                   IFBP  ,ISPP  ,ISFF  ,ISQQ  ,ICQQ  ,ITNM  ,
     2                   IPSF  ,IPSD
      READ  (IRSTRT)     NRFPTS,NRFSDS,NRFLIN,NRFINT,NRFREA,NRFSEL,     NESSUS
     1                   NRFCMP,NRFDOF,NRFSIZ,LRFEND,JRFTYP,JCOUNT,     NESSUS
     2                   JCFLAG,JRFSEL,JRFDOM,NVKEPT,MORE               NESSUS
      READ  (IRSTRT)     IRFPTS,IRFINT,IRFMNR,IRFSDR,IRFSEL,IRFSET,     NESSUS
     1                   IRFMNV,IRFSDV,IRFPTV,IRFVEC,IRFVAL,IRFNRM,     NESSUS
     2                   IRFCOR,IRFID ,IRFWRK                           NESSUS
      READ  (IRSTRT)     IPULSE,KPULSE,IPDTIM,IPDFOR
      READ  (IRSTRT)     MANVAR(   7 ),JPR   ,ICOM1 ,NCOMP
      READ  (IRSTRT)     IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                   ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                   IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      READ  (IRSTRT)     INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      READ  (IRSTRT)     IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                   ISBC  ,ISBCR
      READ  (IRSTRT)     IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                   IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                   IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                   IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                   IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                   IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                   IEQCSI,IPREF ,IDSX3 ,IYIELD,IST441,IST442,
     7                   IST443,IST444,IST445,IST446,IST447,IST448
      READ  (IRSTRT)     IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                   IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                   IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                   IMASDI,IMASUP,IST521,IST522,IST523,IST524
      READ  (IRSTRT)     IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                   IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                   IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                   KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                   KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                   IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                   KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                   IVELM ,IAELM ,IMASEL,KYIELD,IST647,IST648,
     8                   IST649,IST650,IST651,IST652,IST653,IST654
      READ  (IRSTRT)     ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      READ  (IRSTRT)     KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                   IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
```

```
     2                        KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      READ   (IRSTRT)         KEQCSI,KIOMNO,KSWLNO,KTMPNO,KTDFNO,KDUMMY,
     1                        KEQCST,KOMENO,KVSWTO,IST910,IST911,IST912
      READ   (IRSTRT)         ISUBEL,ISUBNP,ISUBPT,NSDATA,ISUBTY,IEMBED
      READ   (IRSTRT)         NSUCRD,NSUNFR,NSUNOD,NSUSTR,NSUCHR,NSUPR ,
     1                        NSUINT,NSULV ,NSUTEM,NSUNDI,NSUSHR,NSUIDF
      READ   (IRSTRT)         NLVSUB(    10),NFRSUB(    10)
      READ   (IRSTRT)         ISHIFT,KSHIFT,IFREQ ,LFREQ ,NOFFST,NFOUND
      READ   (IRSTRT)         TIMINC,TOTINC,RUNTIM
      READ   (IRSTRT)         DALPHA,DBETA ,DGAMMA
      READ   (IRSTRT)         RELERR,ABSERR,REACMX,RESIMX,DISERR,DISTOR,
     1                        ENGTOR,ENGNRM
      READ   (IRSTRT)         ISETUP,MAVAIL,LENREC,NUMREC,LENBLK,NUMBLK,
     1                        IVPAGE,NVPAGE,IVSTRT,IKRO  ,ILCOL ,IPIVCO,
     2                        IHEDER,IFRNRH,IPIVOT,IPIVRO,IPVKOL,IVEND ,
     3                        ISRECD,IERECD
      READ   (IRSTRT)         MAXCOL,NELCOL,LAYPRT,JTENSR
C
C ***********************************************************************
C **  READ BACK THE DATA IN BLANK COMMON STORED IN 64 WORD BLOCKS      **
C ***********************************************************************
C
      NWORDS = 64
      NRECDS = ILAST / NWORDS
C
      DO 600 JREC = 1, NRECDS
        IBEG = 1+(JREC-1)*NWORDS
        IEND = JREC*NWORDS
        READ  (IRSTRT)  ( IWORK(JJ), JJ=IBEG,IEND )
  600 CONTINUE
C
      IBEG = 1+NRECDS*NWORDS
      IEND = ILAST
      IF (IBEG .NE. IEND)
     1  READ  (IRSTRT)  ( IWORK(JJ), JJ=IBEG,IEND )
C
C ***********************************************************************
C **  PRINT MESSAGE AND PROCEED WITH THE ANALYSIS AS USUAL             **
C ***********************************************************************
C
C+    CALL QUIT   ('REST','ART ','FILE',' REA','DER ','    ',1)
      CALL TIMOUT ('REST','ART ','FILE',' WAS',' REA','D   '  )
C
C *** STEP UP THE INCREMENT COUNTER
C
```

```
      JINC = JINC+1
C
      RETURN
      END
```

3.4 Post-Processing Data File

The MHOST program produces an industry-standard formatted post-processing data file when the user requests it. The format is almost compatible with the MARC general purpose finite element program, Versions J.3 and K.1. The difference is that the stress variables (referred to as element variables in most of the finite element programs) are produced at nodal points and written on the file in a tightly packed manner.

CONTENTS OF HOST POST TAPE - Version 4.2

The following describes the contents of the ten blocks which will be found on the Post Tape. This information is the same for either the binary or the formatted Post Tape.

	BLOCK NUMBER	DESCRIPTION
	Block I	TITLE
	NR;80A1;(TITLE(J),J=1,70)	TITLE(J) is Jth character
	Block II	CONTROL INFORMATION
1.	NR;6I3;INUM	Number of variables per element
2.	LNUM	Number of nodes in mesh
3.	MNUM	Number of elements in mesh
4.	NDEG	Number of degrees of freedom per node
5.	NSTRES	Dummy
6.	INOD	Number of nodal variables - See Note below.
7.	IPSTCC	Connectivity and coordinate flag (1- given)

8.	IPSTYP	Type of tape (1- Formatted)
9.	NCRD	Number of coordinates per node
10.	NNODMX	Maximum number of nodes per element
11.	IANTYP	Analysis Type Flag: 2 = Displacement - With Reaction Forces 4 = Dynamics - With Reaction Forces 15 = Eigenvector (Modal) 16 = Eigenvector (Buckle)
12.	ICOMPL	Set to 0 if real analysis Set to 1 if complex analysis
13.	NBCTRA	Number of nodes with transformations
14.	POSTRV	Post tape revision number, 1 for this release (Version 4.2)
	IDM4 IDM5 IDM6 IDM7	Not used; reserved for future expansion

Note:

INOD = {NDEG}*JNODE

```
  If IANTYP =   2   on  2
                4       4
               15       1
               16       1

     IANTYP = 15 only appears during subincrements
     IANTYP = 16 only appears during subincrements
  If IANTYP =  8 and ICOMPL = 1 JNODE = 2
  If IANTYP =  9 and ICOMPL = 1 JNODE = 4
```

Block II CODE NUMBER ASSOCIATED WITH ELEMENT VARIABLES

INUM Records; J=1, INUM

NR;6I3;JPLOT(J)	Element Variable Code
Block IV	CONNECTIVITY LIST
	If IPSTCC is zero this block will be omitted.
	MNUM Records; J=1,MNUM
NR;6I3;LM(1)	Element Type
LM(2)	Number of Nodes in this Element
LM(3)	1st Node of Element J
LM(NNODMX+2)	Last Node of Element J
Block V	COORDINATE LIST
	If IPSTCC is zero this block will be omitted.
	LNUM Records; J=1, LNUM
NR;6E13.6; SUM(1)	1st Coordinate of Jth Node
SUM(2)	
SUM(NCRD)	Last Coordinate of Jth Node
Block VI	TRANSFORMATION LIST
	If NBCTRA is zero this block will be omitted.
	Binary 1 record Formatted (NBCTRA-1)/6 + 1 records
NR;6I13; LM(I)	List of nodes which have transformations applied
Block VII	TRANSFORMATION - DIRECTION COSINES
	If NBCTRA is zero this block will be omitted.

	1 record per each node listed in Block VI if binary tape
NR;6E13.6; D(1,1) D(2,1) D(3,1) D(1,2) D(2,2) D(3,2) D(1,3) D(2,3) D(3,3)	2 records per node if formatted tape Transformations are for local to global

Blocks VIII, IX and X are repeated for each increment.

Block VIII	INCREMENT, TIME AND FREQUENCY
NR;6E13.6 X1(1)	Transient Time
X1(2)	Increment number is a real formed as I + J/100 I is the static increment number J is either the harmonic subincrement or the eigenvector number
X1(3)	Frequency
X1(4)	Flag to read new blocks II, III, IV, V, VI, VII. Set to 1 to read these blocks again.
X1(5)	IANTYP analysis type flag.
X1(6)	Not used; reserved for future expansion.

Block IX	VALUES OF ELEMENT VARIABLES
	If INUM is zero, this block is omitted.
	If IANTYP=15 or IANTYP=16, this block is omitted.
	MNUM*NSTRES Records
NR;6E13.6;	VALUE((CI,J), I=1,INUM), J=1,INODE)
	VALUE is the name of an array storing all the element variables at nodes.

Block X VALUE OF NODAL VARIABLES

If INOD is zero, this block is omitted.

LNUM Records

NR;6E136; SUM(1) SUM(NDEG) SUM(NDEG+1) SUM(2*NDEG) SUM(INOD)	Nodal displacements, velocities, accelerations and reactions

During subincrements:

The first NDEG quantities are:

IANTYP = 15
Nodal Components of Dynamic Mode

IANTYP = 16
Nodal Components of Buckle Eigenvector

NOTES:

NR	Indicates the beginning of a new record for the binary post tape.
NR;Format	Indicates the beginning of a new set of information which is to be read with the following format:

Appendix 1 provides a sample program to dump the post tape.

The table below provides codes for selecting strains and stresses for plotting:

1-6	Components of total strain.
7	Equivalent plastic strain.
8	Equivalent creep strain.
9	Total temperature.
11-16	Component of stress.
17	Equivalent Mises tensile stress.

18	Mean normal stress (tensile positive) (for Mohr-Coulomb).
19	User definable quantity to write on post tape.
20	User definable quantity to write on post tape.
21-26	Physical components of the total plastic strain.
27	Total equivalent plastic strain.
29	Second state variable.
31-36	Physical components of total creep strain.
37	Total equivalent creep strain.

If several layers (shell or beam) are to be plotted, the code number should be as follows: code for variable as above + 100 x layer number.

An example of the post tape file for a 20 four node plane stress element used to mode a cantilever beam is:

```
BEAM PROBLEM
          16          33          20           2           2           4
           1           1           2           4           2           0
           0           1           0           0           0           0
          17
          11
          12
          13
          39
           1
           2
           3
           7
          21
          22
          23
           8
          31
          32
          33
           3           4           1           2           5           4
           3           4           2           3           6           5
           3           4           4           5           8           7
           3           4           5           6           9           8
           3           4           7           8          11          10
           3           4           8           9          12          11
           3           4          10          11          14          13
           3           4          11          12          15          14
           3           4          13          14          17          16
```

	3	4	14	15	18	17
	3	4	16	17	20	19
	3	4	17	18	21	20
	3	4	19	20	23	22
	3	4	20	21	24	23
	3	4	22	23	26	25
	3	4	23	24	27	26
	3	4	25	26	29	28
	3	4	26	27	30	29
	3	4	28	29	32	31
	3	4	29	30	33	32

0.0000	0.50000
0.0000	0.0000
0.0000	-0.50000
2.0000	0.50000
2.0000	0.0000
2.0000	-0.50000
4.0000	0.50000
4.0000	0.0000
4.0000	-0.50000
6.0000	0.50000
6.0000	0.0000
6.0000	-0.50000
8.0000	0.50000
8.0000	0.0000
8.0000	-0.50000
10.000	0.50000
10.000	0.0000
10.000	-0.50000
12.000	0.50000
12.000	0.0000
12.000	-0.50000
14.000	0.50000
14.000	0.0000
14.000	-0.50000
16.000	0.50000
16.000	0.0000
16.000	-0.50000
18.000	0.50000
18.000	0.0000
18.000	-0.50000
20.000	0.50000
20.000	0.0000
20.000	-0.50000

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

6003.0	5999.0	-7.9094	-0.16610E-02	0.61575E-02	0.60014E-02
-0.18076E-02	-0.43187E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.29068E-02	0.51616E-03
-0.96139E-04	-0.16456E-02	0.49706E-08	0.54500E-09	-0.25099E-09	-0.42786E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6003.0	-5999.0	7.9092	-0.16302E-02
0.61575E-02	-0.60014E-02	0.18076E-02	-0.42386E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6003.0	5999.0	-7.9240	-0.29525E-02	0.61575E-02	0.60014E-02
-0.18076E-02	-0.76764E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.52284E-02	-0.30447E-03
0.16348E-03	-0.30093E-02	0.90418E-08	-0.35351E-09	0.25482E-09	-0.78241E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6003.0	-5999.0	7.9244	-0.30660E-02
0.61575E-02	-0.60014E-02	0.18076E-02	-0.79717E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.9	5998.9	-7.9588	-0.34241E-02	0.61574E-02	0.60013E-02
-0.18076E-02	-0.89027E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.59185E-02	-0.50404E-03
-0.36859E-03	-0.34071E-02	0.10247E-07	-0.39346E-09	-0.21738E-09	-0.88584E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.9	-5998.9	7.9579	-0.33900E-02
0.61574E-02	-0.60013E-02	0.18076E-02	-0.88140E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.8	-7.9876	-0.56537E-02	0.61572E-02	0.60012E-02
-0.18076E-02	-0.14700E-07	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.97750E-02	-0.10927E-02
0.17802E-03	-0.56015E-02	0.16856E-07	-0.11461E-08	0.50583E-09	-0.14564E-07
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.8	-5998.8	7.9881	-0.55494E-02
0.61572E-02	-0.60012E-02	0.18076E-02	-0.14428E-07	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.8	-7.9996	0.37059E-02	0.61572E-02	0.60012E-02
-0.18076E-02	0.96353E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.61729E-02	-0.10849E-02
-0.88108E-03	0.35170E-02	0.10649E-07	-0.82053E-09	-0.55562E-09	0.91441E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.8	-5998.8	7.9979	0.33281E-02
0.61572E-02	-0.60012E-02	0.18076E-02	0.86529E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.8	-7.9876	0.17084E-02	0.61573E-02	0.60012E-02
-0.18076E-02	0.44419E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.33956E-02	0.26616E-03
0.93227E-03	0.19007E-02	0.57894E-08	-0.13521E-10	0.85242E-09	0.49419E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	0.0000	6002.8	-5998.8	7.9891	0.20930E-02
0.61573E-02	-0.60012E-02	0.18076E-02	0.54419E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.9	-7.9647	0.67373E-02	0.61573E-02	0.60012E-02
-0.18076E-02	0.17517E-07	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.11727E-01	-0.89912E-03
-0.16063E-02	0.67225E-02	0.20266E-07	-0.41724E-09	-0.13365E-08	0.17479E-07
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.8	-5998.9	7.9620	0.67078E-02
0.61573E-02	-0.60012E-02	0.18076E-02	0.17440E-07	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.9	5998.9	-7.9466	0.30772E-02	0.61574E-02	0.60013E-02
-0.18076E-02	0.80008E-08	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.70974E-02	0.46497E-02
0.17720E-02	0.33592E-02	0.11251E-07	0.41181E-08	0.37708E-09	0.87338E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.9	-5998.9	7.9494	0.36411E-02
0.61574E-02	-0.60013E-02	0.18076E-02	0.94669E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.9	-7.9636	-0.40192E-02	0.61573E-02	0.60012E-02
-0.18076E-02	-0.10450E-07	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.84715E-02	-0.10157E-02
-0.18358E-02	-0.48038E-02	0.14573E-07	-0.46498E-09	-0.15311E-08	-0.12490E-07
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.8	-5998.8	7.9616	-0.55885E-02
0.61573E-02	-0.60012E-02	0.18076E-02	-0.14530E-07	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.8	5998.7	-7.9986	-0.14157E-03	0.61572E-02	0.60011E-02
-0.18076E-02	-0.36807E-09	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.14448E-02	0.54107E-03
0.10165E-02	-0.66112E-03	0.22925E-08	0.23611E-09	0.85422E-09	-0.17189E-08
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.8	-5998.8	7.9986	-0.11807E-02
0.61572E-02	-0.60012E-02	0.18076E-02	-0.30697E-08	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
6002.7	5998.7	-8.0174	-0.12263E-01	0.61572E-02	0.60011E-02
-0.18076E-02	-0.31883E-07	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.20265E-01	0.10427E-01
0.36572E-02	-0.10436E-01	0.33235E-07	0.93296E-08	0.52916E-09	-0.27133E-07
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	6002.7	-5998.7	8.0146	-0.86084E-02
0.61572E-02	-0.60011E-02	0.18076E-02	-0.22382E-07	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.95018E-27	-0.90381E-03	-999.84	-3.9566		
0.24995E-33	-0.19943E-26	-0.80276E-04	7.9155		

```
-0.95018E-27 -0.90381E-03   999.84       -3.9566
 0.12003E-01 -0.24909E-01  0.41666E-02  -7.9271
 0.10900E-08 -0.24006E-01  0.23057E-03   15.855
-0.12003E-01 -0.24909E-01 -0.38388E-02  -7.9273
 0.24005E-01 -0.96926E-01  0.10585E-01  -7.9577
-0.14141E-08 -0.96022E-01  0.20487E-03   15.916
-0.24005E-01 -0.96926E-01 -0.10416E-01  -7.9569
 0.36008E-01 -0.21695      0.58989E-02  -7.9858
-0.48385E-09 -0.21605      0.82186E-04   15.968
-0.36008E-01 -0.21695     -0.57784E-02  -7.9855
 0.48010E-01 -0.38499     -0.68275E-04  -7.9970
-0.59985E-08 -0.38409     -0.33751E-03   15.989
-0.48010E-01 -0.38499     -0.66963E-04  -7.9954
 0.60013E-01 -0.60103     -0.11629E-02  -7.9860
-0.37660E-08 -0.60013     -0.21884E-04   15.971
-0.60013E-01 -0.60103      0.89412E-03  -7.9867
 0.72015E-01 -0.86509      0.88963E-04  -7.9663
-0.60526E-08 -0.86419     -0.81361E-03   15.930
-0.72015E-01 -0.86509     -0.12875E-02  -7.9640
 0.84018E-01  -1.1772      0.47992E-02  -7.9504
-0.54349E-08  -1.1763     -0.24978E-03   15.909
-0.84018E-01  -1.1772     -0.49747E-02  -7.9526
 0.96020E-01  -1.5372      0.10184E-01  -7.9669
 0.10420E-07  -1.5363      0.28485E-03   15.933
-0.96020E-01  -1.5372     -0.79310E-02  -7.9644
 0.10802      -1.9453      0.26281E-02  -7.9940
-0.72949E-08  -1.9444     -0.12270E-02   15.992
-0.10802      -1.9453     -0.22268E-02  -7.9956
 0.12002      -2.4014     -0.21691      -4.0056
 0.11364E-07  -2.4005      0.17279E-02   8.0074
-0.12002      -2.4014      0.21566      -4.0073
```

The write operation takes place in SUBROUTINE POSTOU:

```
C=SUBROUTINE=POSTOU  CALLED BY SECOND LEVEL DRIVER ROUTINES
      SUBROUTINE POSTOU
     1     ( IWORK , RWORK , ISIZE )
C
C ************************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C ************************************************************************
```

```
C
      DIMENSION IWORK( ISIZE ) , RWORK( ISIZE )
      DIMENSION JPLOT(    80 )
      DIMENSION KPLOT(    80 )
      DIMENSION BUFFER(   50 ) , VALUE(    50 )
      DIMENSION LOCATE(   10 )
C
C ************************************************************************
C
C     COMMON BLOCKS IN ALPHABETICAL ORDER
C
C ************************************************************************
C
      COMMON / ALGEM  / ICREAD,ILPRNT,JLPRNT,ICONSL,IPOSTF,ISCRAF,
     1                  IPLOTB,IRSTRT,JCREAD,IPVARS,IPSETS,IFILEX,
     2                  PI    ,LINE  ,LINE2
      COMMON / PARAM  / NTYPE ,NELEM ,NNODE ,NBC   ,NTIE  ,NMAX  ,
     1                  NTRAN ,NTRAC ,NFD   ,NBAND ,NEXT  ,NSUB  ,
     2                  NPRINT,NPOST ,NSBC  ,NDUP  ,NSIZE ,NBSECT,
     3                  NSHIFT,NSBFGS,NGMRS ,NSPRI ,NMASS ,NDASH ,
     4                  NDYNMD,NSBNC ,NSUPER,NHARM ,NBASE ,NINC  ,
     5                  NITER ,NPSPTS,NFDPTS,NPULSE,NPDPTS,NHARD ,
     6                  NSUMCH,NDIMEN,NMONIT,NPAR40,NPAR41,NPAR42,
     7                  NPAR43,NPAR44,NPAR45,NPAR46,NPAR47,NPAR48
      COMMON / CTITLE / TITLE (   20),IDAT  (    5),IDATE2,ICLOCK,
     1                  IFCRAY
      COMMON / EIGEN  / IEGNVC,IGNMS ,IOMEG ,IMOENO,IDYNMD,ISTRT2,
     1                  IPTAR ,IPTBR ,IPTVED,IMDAM ,IOMEGD
      COMMON / ELEMEN / IC    ,IEL   ,IDF   ,JLAW  ,IPATH ,IASSEM,
     1                  JRULE ,JCART ,JEL009,JEL010,JEL011,JEL012
      COMMON / ELTYP  / NELCRD,NELNFR,NELNOD,NELSTR,NELCHR,NELPR ,
     1                  NELINT,NELLV ,NELLAY,NDI   ,NSHEAR,NELCMP
      COMMON / ERRORS / IERR
      COMMON / MACHIN / IDP
      COMMON / MAXIMA / MAXCRD,MAXNFR,MAXNOD,MAXSTR,MAXCHR,MAXPRS,
     1                  MAXLAY,MAXINT,MAXWRK,MAXNLV,NSUMAX,MAXCMP,
     2                  MAXBSP,MAXGMR,MAXTEM,MAXELM,MAXLWK,MAXDMT,
     3                  MAXFRN,MAXBET,MAXVAR,MAXSET,MAXEAN,MAXORD,
     4                  MAX025,MAX026,MAX027,MAX028,MAX029,MAX030
      COMMON / CONTRO / JEND  ,JITER ,JTEMP ,JPRINT,JP    ,JSUB  ,
     1                  JINC  ,JREST ,JSAVE ,JREDIM,JAUTO ,JPOST ,
     2                  JBACK ,JOPTIM,JCREEP,JDIST ,JCONST,JDYN  ,
     3                  NONISO,ITHERM,ITRIG ,IDYNM ,JREPOT,JTANGE,
     4                  JTHERM,JFORCE,JUTEMP,JUCOEF,JDISTS,JUHOOK,
     5                  JDERIV,JUBOUN,IDSTOP,INTSTR,JPLAST,JBAND ,
```

```
     6                      JFRONT,JDEFOR,JEMBED,ITEST ,JDISP ,IFBFGS,
     7                      IFSCNT,IFLINE,IFPRNT,ICOMPS,IPCONJ,JEIGEN,
     8                      IFBODY,IFGRAV,IFCENT,JDAMP ,LDYN  ,ISTAT ,
     9                      JFDSXX,JISTIF,JCENTM,JFINIT,JLARGE,JFOLOW,
     +                      JWKSLP,JPRES ,JCDUM2,JCDUM3
      COMMON / START1 / IELPRM,ITYP  ,INEL  ,ICHAR ,IPRES ,ISTRS ,
     1                  ISTRN ,ICOP  ,IPRINT,IPOST ,IDIST ,ILEAN ,
     2                  IBPRES,IBNORM,IMONIT,IST116,IST117,IST118
      COMMON / START2 / INOD  ,ITEM  ,INLV  ,IPOSU ,ITEMDF,IDUP
      COMMON / START3 / IKBC  ,ITI   ,ITR   ,ITRAN ,ITRAC ,IEXT  ,
     1                  ISBC  ,ISBCR
      COMMON / START4 / IDINC ,IDTOT ,IFORCE,IRESID,IWTNOD,ISIGNO,
     1                  IEPSNO,IPSTRN,ICSTRN,ITSTRN,IISTRS,IISTRN,
     2                  IIPSTR,IICSTR,IITSTR,IPSTNO,ICSTNO,ITSTNO,
     3                  IISTNO,IISNNO,IIPSNO,IICSNO,IITSNO,IDMAT ,
     4                  IDMTNO,IEQCST,IOMENO,IIOMNO,ITDSNO,IVSWTO,
     5                  IDYNV ,IDYNA ,IDSX1 ,IDSX2 ,IDSITR,ISWELL,
     6                  IEQCSI,IPREF ,IDSX3 ,IYIELD,IST441,IST442,
     7                  IST443,IST444,IST445,IST446,IST447,IST448
      COMMON / START5 / IRL   ,IREAC ,IES   ,IAB   ,IBQM  ,ISRL  ,
     1                  IBTLC ,ISKM  ,ILAST ,IRLB  ,IDINCP,IFORIN,
     2                  IOP   ,IDAM  ,IMASMT,IDIAG ,IUPTRI,ICOLPT,
     3                  IMASDI,IMASUP,IST521,IST522,IST523,IST524
      COMMON / START6 / IELV  ,ICOR  ,ISIG  ,IEPS  ,IWNOD ,ISNOD ,
     1                  IENOD ,IETM  ,ICH   ,IPP   ,IXRL  ,IXIRL ,
     2                  IXP   ,IXK   ,KPSTRN,KCSTRN,KTSTRN,KISTRS,
     3                  KISTRN,KIPSTR,KICSTR,KITSTR,KPSTNO,KCSTNO,
     4                  KTSTNO,KISTNO,KISNNO,KIPSNO,KICSNO,KITSNO,
     5                  IMASNO,IMNOD ,IEQPST,IEQPSI,KEQPST,KEQPSI,
     6                  KDMAT ,KDMTNO,KTDSNO,KITDST,IXM   ,IXC   ,
     7                  IVELM ,IAELM ,IMASEL,IST646,IST647,IST648,
     8                  IST649,IST650,IST651,IST652,IST653,IST654
      COMMON / START7 / ICON  ,IKBCR ,ITRACR,ITRANR,IBETA ,IDET
      COMMON / START8 / KGEPS ,KIGEPS,KGSIG ,KIGSIG,KGTDST,
     1                  IGEPNO,IIGENO,IGSINO,IIGSNO,IGTDNO,
     2                  KGEPNO,KIGENO,KGSINO,KIGSNO,KGTDNO
      COMMON / TIME   / TIMINC,TOTINC,RUNTIM
C
C     ---------------------------------------------------------------
C     IPOSTF ( 19 )    I  FORMATTED OUTPUT FILE FOR POST-PROCESSING
C     ---------------------------------------------------------------
C      VERSION 2.0 GENERATES MARC COMPATIBLE POST TAPE CONTAINING
C      ONLY MESH DATA AND DISPLACEMENT + REACTION AT NODES.
C      THIS CAN BE PROCESSED BY INVOKING 'POSTDATA' COMMOAND WITH 'MARC'
C      OPTION IN 'MENTAT' INTERACTIVE SESSION.
```

```
C
C ********************************************************************
C
C     FULLY MENTAT COMPATIBLE POST PROCESSING FILE WILL EVENTUALLY
C     WRITTEN ON IPLOTF FILE AND THEN IPOSTF WILL BE USED FOR THE
C     TOPOLOGICAL MESH DEFINITION OUTPUT TO BE PROCESSED BY 'READ'
C     OPTION IN MENTAT
C
C ********************************************************************
C
                                  MINCMP = MAXCMP
      IF( MAXCMP .EQ. 8 ) MINCMP = 5
C
      IZERO        =   0
      MAXVAL       =  50
      MAXQNT       =   4
      LOCATE( 1 ) = ISIGNO
      LOCATE( 2 ) = IEPSNO
      LOCATE( 3 ) = IPSTNO
      LOCATE( 4 ) = ICSTNO
C
C --- HEADER RECORDS ARE WRITTEN IN THE POST TAPE ONLY WHEN THIS
C --- ROUTINE IS ENTERED AT THE ZERO-TH INCREMENT. OTHERWISE THE NODAL
C --- VALUES ARE WRITTEN IN THE POST TAPE.
C
      IF(  JINC .GT.  0    )    GO   TO   9000
C
C --- FIRST RECORD IN THE POST TAPE : TITLE ---------------------------
C
      WRITE(IPOSTF,1000) TITLE
 1000 FORMAT(20A4)
C
C --- SECOND RECORD : CONTROL INFORMATION -----------------------------
C
C     SET VARIABLES ACCORDING TO THE MARC DOCUMENT
C
      INUM     = ( MINCMP + 1 ) * 4
      IF( JEIGEN .EQ. 1 ) INUM   = 0
      LNUM     =   NNODE
      MNUM     =   NELEM
      NDEG     =   MAXNFR
      NSTRES   =     2
      JNOD     =   MAXNFR * 2
      IPSTCC   =     1
      IPSTYP   =     1
```

```
      NCRD      =    MAXCRD
      IF( NCRD    .EQ. 7 ) NCRD    =    3
      NNODMX    =    NELNOD
      IANTYP    =      2
      IF( JEIGEN .EQ. 1 .AND. IDYNM .EQ. 1 ) IANTYP =  15
      IF( JEIGEN .EQ. 1 .AND. IDYNM .EQ. 0 ) IANTYP =  16
      ICOMPL    =      0
      NBCTRA    =      0
      IPOSTR    =      1
      IDM4      =      0
      IDM5      =      0
      IDM6      =      0
      IDM7      =      0
C
      WRITE(IPOSTF,1010) INUM   ,LNUM   ,MNUM   ,NDEG   ,NSTRES,JNOD   ,
     1                   IPSTCC,IPSTYP,NCRD   ,NNODMX,IANTYP,ICOMPL,
     2                   NBCTRA,IPOSTR,IDM4   ,IDM5   ,IDM6   ,IDM7
 1010 FORMAT(6I13)
C
C --- CODE NUMBER ASSOCIATED WITH ELEMENT VARIABLES -------------------
C
      IF( JEIGEN .EQ. 1 ) GO TO 3000
C
                         JPLOT(    1     )    =    17
                         JPLOT(    2     )    =    11
                         JPLOT(    3     )    =    12
                         JPLOT(    4     )    =    13
      IF(MINCMP .GE. 4 ) JPLOT(    5     )    =    14
      IF(MINCMP .GE. 5 ) JPLOT(    6     )    =    15
      IF(MINCMP .GE. 6 ) JPLOT(    7     )    =    16
                         INDX                 =                MINCMP + 1
                         JPLOT( INDX + 1 )=    39
                         JPLOT( INDX + 2 )=    1
                         JPLOT( INDX + 3 )=    2
                         JPLOT( INDX + 4 )=    3
      IF(MINCMP .GE. 4 ) JPLOT( INDX + 5 )=    4
      IF(MINCMP .GE. 5 ) JPLOT( INDX + 6 )=    5
      IF(MINCMP .GE. 6 ) JPLOT( INDX + 7 )=    6
                         INDX                 =    INDX     +  MINCMP + 1
                         JPLOT( INDX + 1 )=    7
                         JPLOT( INDX + 2 )=   21
                         JPLOT( INDX + 3 )=   22
                         JPLOT( INDX + 4 )=   23
      IF(MINCMP .GE. 4 ) JPLOT( INDX + 5 )=   24
      IF(MINCMP .GE. 5 ) JPLOT( INDX + 6 )=   25
```

```
      IF(MINCMP .GE. 6 ) JPLOT( INDX + 7 )=   26
                         INDX             =    INDX     +   MINCMP + 1
                         JPLOT( INDX + 1 )=    8
                         JPLOT( INDX + 2 )=   31
                         JPLOT( INDX + 3 )=   32
                         JPLOT( INDX + 4 )=   33
      IF(MINCMP .GE. 4 ) JPLOT( INDX + 5 )=   34
      IF(MINCMP .GE. 5 ) JPLOT( INDX + 6 )=   35
      IF(MINCMP .GE. 6 ) JPLOT( INDX + 7 )=   36
                         INDX             =    INDX     +   MINCMP + 1
C
      WRITE(IPOSTF,1030) (JPLOT(K),K=1,INDX)
 1030 FORMAT(I13)
C
 3000 CONTINUE
C
C --- DATA BLOCK IV : ELEMENT CONNECTIVITY DATA -----------------------
C
      DO      6000     IEL      =     1     ,   NELEM
C
                       IC       = IWORK( ITYP + IEL   - 1 )
                       NELNOD   = 0
C
                       IF( IC     .EQ. 0 ) GO TO 6007
C
      CALL ELVULV
     1     (IWORK ,IC     ,IERR   )
C
                       IADRES   =  INEL   +  MAXNOD * ( IEL    - 1 )
                       JADRES   = IADRES +  NELNOD - 1
C
C ... SPECIAL TRICK ASSOCIATED WITH THE FRONTAL SOLUTION
C
      DO 6005 INTPNT =    1   , NELNOD
              JNTPNT = INTPNT + IADRES - 1
 6005   KPLOT(INTPNT)= IABS(IWORK(JNTPNT))
C
 6007                  CONTINUE
C
C ... REPLACE THE ASSUMED STRAIN ELEMENTS WITH THE MATCHING MARC/MENTAT
C     ELEMENT NUMBERS  (A LITTLE INNOCENT LIE THAT WON'T HURT ANYBODY)
C
                         IC1 = IC
      IF ( IC .EQ. 151 ) IC1 =  3
      IF ( IC .EQ. 152 ) IC1 = 11
```

```
      IF ( IC .EQ. 153 ) IC1 = 10
      IF ( IC .EQ. 154 ) IC1 =  7
C
      WRITE(IPOSTF,1010) IC1    ,NELNOD,(KPLOT(K),K=1,NELNOD)
C
 6000                   C  O  N  T  I  N  U  E
C
C --- DATA BLOCK V : NODAL COORDINATE DATA -----------------------------
C
      DO       6010      INODE    =      1      ,    NNODE
C
                         IADRES   =   INOD + MAXCRD * ( INODE - 1 ) * IDP
      CALL POSTPR
     1     (RWORK(IADRES),NCRD,IPOSTF)
C
 6010                   C  O  N  T  I  N  U  E
 9000 C O N T I N U E
C
C --- INCREMENTAL HEADER RECORD ----------------------------------------
C
      NSUBIN     =      1
      IF( JEIGEN .EQ. 1 ) NSUBIN = NDYNMD
C
      DO 8000 ISUBIN = 1, NSUBIN
C
      RINCRE     =     JINC
      FREQUN     =      0
      RFLAGS     =      0
      RANTYP     =      0
      RDUMMY     =      0
C
      IF( JEIGEN .EQ. 0 ) GO TO 5100
C
      RINCRE = RINCRE + 0.01D0 * ISUBIN
      INDX   = IOMEG  + ( ISUBIN - 1 ) * IDP
      CALL COPY( RWORK(INDX) , FREQUN , 1 )
      IF( IDYNM .EQ. 1 ) RANTYP = 15
      IF( IDYNM .EQ. 0 ) RANTYP = 16
C
 5100 CONTINUE
C
      WRITE(IPOSTF,1020) RUNTIM,RINCRE,FREQUN,RFLAGS,RANTYP,RDUMMY
 1020 FORMAT(6G13.6)
C
      IF( JEIGEN .EQ. 1 ) GO TO 7500
```

```
C
                              INUM        = ( MINCMP + 1 ) * 4
                              INITAL      =   0
      DO        7000          INODE       =   1     ,      NNODE
C
                              IREC        =   0
C
      CALL NUL( VALUE , MAXVAL )
C
C *** VALUES CALCULATED AT THE LAYER LEVEL *****************************
C
                            INTLAY    =     1
      IF( NELLAY .NE. 1 )   INTLAY    =     3
C
C *** FIRST DATA OF THE SERIES IS TEMPERATURE *************************
C
C+    LOCTEM = ITEM + (( INODE - 1 )*MAXLAY + INTLAY - 1 )*IDP
C+    IREC   = 1
C
C+    CALL COPY( RWORK(LOCTEM) , VALUE(IREC) , 1 )
C
C *** TENSORIAL QUANTITIES AND THEIR INVARIANTS ***********************
C
C     (I)     TOTAL STRESS
C     (II)    TOTAL STRAIN
C     (III)   PLASTIC STRAIN
C     (IV)    CREEP STRAIN
C
      DO 5200 INTQNT = 1 ,MAXQNT
C
      INTLOC = LOCATE(INTQNT) + (( INODE - 1 )*MAXLAY + INTLAY - 1 )
     1                         * IDP * MAXCMP
C
      CALL COPY( RWORK(INTLOC) , BUFFER , MINCMP )
      IF( INTQNT .EQ. 1 ) CALL EQVS
     1  ( EQVAL , BUFFER , MINCMP , NDI , NSHEAR , JLAW ,  NONISO ,
     2     INODE )
      IF( INTQNT .GT. 1 ) CALL EQVC
     1  ( EQVAL , BUFFER , MINCMP , NDI , NSHEAR , JLAW )
C
                                             IREC = IREC + 1
      CALL COPY( EQVAL , VALUE(IREC) , 1 )
                                             IREC = IREC + 1
      CALL COPY( BUFFER , VALUE(IREC) , MINCMP )
                                             IREC = IREC + MINCMP - 1
```

```
 5200 CONTINUE
C
      CALL POSTEN( VALUE , INUM , IPOSTF , INODE , NNODE , INITAL )
C
 7000                 C O N T I N U E
C
 7500 CONTINUE
C
C --- TOTAL DISPLACEMENT AT THE CURRENT INCREMENT AND REACTION FORCE ---
C
      IF(NTRAN .EQ. 0)          GO TO 9550
C
           CALL TRANS2(NTRAN,RWORK(ITRANR),IWORK(ITRAN),RWORK(IDTOT),
     1                 NFD,-1.D0,NELNFR,JLAW)
           CALL TRANS2(NTRAN,RWORK(ITRANR),IWORK(ITRAN),RWORK(IDINC),
     1                 NFD,-1.D0,NELNFR,JLAW)
 9550                           CONTINUE
C
                         IDVEC  = IDTOT
      IF( JEIGEN .EQ. 1 ) IDVEC  = IEGNVC + ( ISUBIN - 1 ) * NFD * IDP
C
      DO        5000          INODE      =   1    ,      NNODE
C
C ... TOTAL DISPLACEMENTS OR EIGENVECTOR
C
      JNTLOC = INLV   + ( INODE - 1 ) * MAXNFR
      KNTLOC = IDVEC  + ( IWORK(JNTLOC) - 1 )*IDP
      CALL COPY( RWORK(KNTLOC) , VALUE          , MAXNFR )
      NENTRY = MAXNFR
C
C ... NODAL REACTION VECTOR
C
      INDX   = NELNFR + 1
      LNTLOC = IRESID + ( IWORK(JNTLOC) - 1 )*IDP
      CALL COPY( RWORK(LNTLOC) , VALUE( INDX ) , MAXNFR )
      NENTRY = NENTRY + MAXNFR
C
      CALL POSTPR
     1    (  VALUE , NENTRY , IPOSTF   )
C
 5000                 C O N T I N U E
C
      IF(NTRAN .EQ. 0)          GO TO 9530
C
           CALL TRANS2(NTRAN,RWORK(ITRANR),IWORK(ITRAN),RWORK(IDTOT),
```

```
     1                   NFD,1.D0,NELNFR,JLAW)
            CALL TRANS2(NTRAN,RWORK(ITRANR),IWORK(ITRAN),RWORK(IDINC),
     1                   NFD,1.D0,NELNFR,JLAW)
 9530                              CONTINUE
C
 8000 CONTINUE
C
      R     E     T     U     R     N
      END
```

The data record packing for the element data is carried out in the following SUBROUTINE POSTEN:

```
C=SUBROUTINE=POSTEN  POST FILE GENERATION UTILITY
      SUBROUTINE POSTEN
     1    (ARRAY ,NENTRY,IPOSTF,INODE ,NNODE ,INITAL)
C
C ***********************************************************************
C
      IMPLICIT REAL*8 ( A-H , O-Z )
               REAL*4   RWORK
C
C ***********************************************************************
C
      DIMENSION ARRAY (NENTRY)
C
      COMMON / POSTPN / IPOINT,JPOINT,KPOINT,NDATA ,PRNTBF(   6   )
C
      IF( INITAL .NE. 0 ) GO TO 4950
                                         IPOINT =  1
                                         JPOINT =  1
                                         KPOINT =  1
                                         NDATA  =  6
                                         INITAL =  1
 4950                    CONTINUE
C
C     IPOINT : POINTER FOR THE LAST ENTRY OF THE PRINT BUFFER
C     JPOINT : POINTER FOR THE CURRENT ARRAY ENTRY TO BE PRINTED
C
      JPOINT   =   1
C
 5000                C  O  N  T  I  N  U  E
C
      KPOINT = JPOINT + 5
C
```

```
      IF( KPOINT .GT. NENTRY ) GO TO 6000
C
      CALL COPY  ( ARRAY(JPOINT)    , PRNTBF(IPOINT) , NDATA  )
      CALL POSTPR( PRNTBF           ,    6           , IPOSTF )
C
      JPOINT = JPOINT + NDATA
      NDATA  =   6
      IPOINT =   1
C
      G O   T O   5 0 0 0
C
C --- HALF FILL THE PRINTER BUFFER AND RETURN -------------------------
C
 6000                C O N T I N U E
C
      MDATA  = NENTRY - JPOINT + 1
      NDATA  = 6 - MDATA
      IPOINT = MDATA + 1
C
      CALL COPY  ( ARRAY(JPOINT) , PRNTBF(   1  ) , MDATA  )
C
C --- IF THE LOAST NODE IS ENTERED WRITE THE LAST RECORD EVEN IF IT IS
C --- INCOMPLETE
C
      IF( INODE .LT. NNODE ) GO TO 9000
      IF( MDATA .EQ.   0   ) GO TO 9000
C
      CALL NUL   ( PRNTBF(IPOINT) , NDATA )
      CALL POSTPR( PRNTBF         ,   6   , IPOSTF )
C
C ***********************************************************************
C
 9000 RETURN
      END
```

APPENDIX SUBROUTINES

Subroutines included in the MHOST code, Version 4.2, are summarized in this appendix. The routine names are sorted in alphabetical order. A brief description of each routine is given, in conjunction with the names of common blocks referenced therein. Names of routines referenced in some of the vitally important routines are given in this document. Note that almost all subprograms written for the MHOST package are self-documented and further information can be obtained directly from the source listing.

ACCLIN

Reads in initial acceleration data from the main input data reader.

Common block: MACHIN

ADAPTC

Controls the adaptive time step size adjustment for creep strain evaluation

Common block: ALGEM

ADAPTD

Controls the adaptive time step size adjustment for transient dynamic calculations. Not fully functional in Version 4.2.

Common block: TIME, TOLER, CONTRO

ADAPTS

Controls the adaptive load increment size adjustment for the arc length method

Common block: ALGEM

ADD

Adds two double precision real vectors.

ADDBAN

Adds the lumped values listed in SVAL with the connectivity specified in LCON to the global array GABF in band form. Not used for profile-stored global arrays.

ADDIAG

Adds the lumped values listed in SVAL to the global array. This subroutine is the profile-storage counterpart of ADDBAN.

ADDINC

Updates total quantities by adding together the values at the beginning of the increment and the incremental values.

Common block: AUTOIN, ELTTP, CONTRO, SUBELM, MAXIMA, PARAM, START1, START2, START3, START4, START6, START8, TIME

ASMVEC

Assembles global vector from the d.o.f. conversion table. See also SUBST1.

ASSEM1

Assembles the displacement stiffness matrix for quasi-static analysis. Works with the profile solver.

Common block: ALGEM, ADDVAL, AUTOIN, BSECT, CONTROL, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, PULSES, START1, START2, START3, START4, START5, START6, START7, START8, SUBELM, SUBSTR, TIME, TRANSF, MACHIN

ASSEM2

Assembles the global time integration operator matrix for direct time integration. Works with the profile solver.

Common block: ALGEM, ADDVAL, AUTOIN, BSECT, CONTROL, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, PULSES, START1, START2, START3, START4, START5, START6, START7, START8, SUBELM, SUBSTR, TIME, TRANSF, MACHIN

ASSEM3

Assembles the displacement stiffness matrix for quasi-static analysis. Works with the frontal solution.

Common block: ALGEM, ADDVAL, AUTOIN, BSECT, CONTROL, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, PULSES, START1, START2, START3, START4, START5, START6, START7, START8, SUBELM, SUBSTR, TIME, TRANSF, MACHIN

ASSEM4

Assembles the residual vector for quasi-static and transient dynamic analyses. The routine includes the variational recovery of nodal strains, nodal stress recovery and element loop for the internal force calculation.

Common block: ALGEM, ADDVAL, BSECT, COUNTR, CONTRO, ELEMEN, ELTYP, ERRORS,

MACHIN, MAXIMA, SUBELM, SUBTYP, LOUBIN, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, TRANSF, TIME

ATTRIB

Attributes the element connectivity data by the duplicate node and tying options.

Common block: ALGEM, ERRORS, MAXIMA

BACSUB

Performs back-substitution for the nodal displacement vector as a part of the frontal solution subsystem.

BANDBR

Calculates the maximum bandwidth.

Common block: ELTYP

BANNER

Prints banner on the line printer output file.

BASAXS

Generates strain-displacement matrix for axisymmetric element based on the assumed stress method.

BASEIN

Reads in parameters defining harmonic base excitation.

Common block: MACHIN

BASPSN

Generates strain-displacement matrix for plane strain element derived from the assumed stress method.

BASPST

Generates strain-displacement matrix for plane stress element derived from the assumed stress method.

BAXSYM

Generates strain-displacement matrix for axisymmetric element by the numerical integration.

BEAMIN

Reads in beam section properties.

Common block: MACHIN

BFGSLH

Performs the BFGS update for the left hand side of the nonlinear equations.

BFGSRH

Performs the BFGS update for the right hand side of the nonlinear equations.

BFGSVW

Generates the global projection vectors required for the BFGS update.

BFLOAD

Controls the generation of element body force load vector.

BMSTRS

Evaluates constitutive equations for the beam element. Note that the current version only supports a linear elastic beam element.

BNDTRM

Preintegrates the lamina constitutive matrix to obtain the constitutive resultant with respect to the bending moment - curvature terms for shell elements.

BODYIN

Reads in parameters defining body force loading.

Common block: MACHIN

BOUND1

Applies the displacement constraint by eliminating row and column of the global profile-stored stiffness equations.

BOUND2

Applies the displacement constraint to a global vector.

BOUND3

Applies the displacement constraint to a substructure. Obsolete in Version 4.2.

BOUNDN

Calculates the weighted normal vector at nodes on the boundary.

Common block: ALGEM

BOUNFR

Applies the nodal displacement constraint to the frontal solution elimination process.

BOUNIN

Reads in boundary displacement constraint data.

Common block: MACHIN

BPSTRN

Generates strain-displacement matrix for isoparametric plane strain element by numerical integration.

BPSTRS

Generates strain-displacement matrix for isoparametric plane stress element by numerical integration.

BREAD

Reads an array from a specified Fortran I/O unit stored in binary form.

BSHELL

Generates strain-displacement matrix in resultant form for isoparametric shell element by means of selective reduced integration.

BSOLID

Generates strain-displacement matrix for isoparametric solid element by numerical integration.

BTBEAM

Generates strain-displacement matrix for the linear isoparametric Timoshenko beam element by reduced integration.

BUCKLE

Drives the buckling analysis using subspace iterations for eigenvalue extraction.

Common block: AUTOIN, MAXIMA, CTITLE, DAMP, EIGEN, INCCON, MACHIN, ELEMEN, ELTYP, CONTRO, SUBSTR, ERRORS, PARAM, START1, START2, START3, START4, START5, START6, START8, TIME

BULKIN

Supervises the reading operation of the bulk (model) data.

Common block: CONTRO PARAM, SUBELM

External reference: INITI1, DATIN2, DATOU1, CHKELM, SUBDIV

BWRITE

Writes an array to a specified Fortran I/O unit in binary form.

CENMAS

Assembles the stiffness matrix entries related to the centrifugal mass terms for an element.

Common block: BODYFR

External reference: NUL, GAUSSP, COPY, S2DO4N, VSH04N, TBM02N, S1DOZN, S3DO8N, D2DO4N, D3DO8N, DAXO4N, GSHO4N, DBMO2N, SUBT

CENT2D

Calculates the centrifugal load vector for a two-dimensional (plane stress/strain) element.

CENT3D

Calculates the centrifugal load vector for a three-dimensional solid element.

CENTAX

Calculates the centrifugal load vector for a axisymmetric element.

CENTBM

Calculates the centrifugal load vector for a beam element.

CENTSH

Calculates the centrifugal load vector for a shell element.

CHARIN

Reads in material property data.

Common block: MACHIN, ELTYP

CHCHAR

Identifies the value of a number given as a single character.

CHKELM

Checks to see if the element connectivity is given in the counter

clockwise manner. If not, prints warning messages and repairs the connectivity table entries. Dangerous.

Common block: ALGEM, AUTOIN, BSECT, CONTRO, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, SUBSTR, TIME, TRANSF, MACHIN

External reference: ELVULV, QNODEL, INSIDE, LINES

CNODEL

Pulls out information related to the current element. See Section 2.2 for detail.

Common block: TMARCH, ALGEM, CONTRO, ELTYP, MAXIMA, PARAM, START1, START2, START4, START5, START6, START7, START8, START9, MACHIN

External reference: SEARC1, SEARCH, INTERP, ROTDMT, TSHO4N, SHTRAN, TRANSP, MATINV, TBM02N, ADD, ADDSMU, NUL

CNSMAS

Assembles the consistent mass matrix for an element.

CNSTNM

Constructs a consistent transformation from nodal to modal data for linear dynamics by modal superposition.

Common block: EIGEN, ELTYP, MACHIN, MAXIMA, MODSUP, PARAM, POWER, START1, START2, START6

COLRED

Performs column-wise reduction for back-substitution.

COMPDF

Copies the definition of element parameters in the workspace for composite shells

Common block: START1

COMPIN

Reads material property data for the composite laminate option with shell elements.

Common block: ALGEM

COMPRO

Computes the profile column heights and pointers for profile-storage of the global stiffness array.

Common block: ELTYP, MAXIMA, PARAM, ALGEM

CONDSE

Removes specified row and column of a square matrix.

CONNIN

Reads element connectivity data.

Common block: ELTYP

CONTIN

Adds incremental load (defined as the nodal force) to the total load vector.

Common block: MACHIN

COORIN

Reads coordinate data for nodes.

COPY

Copies a double precision real array to another double precision real array.

COPYDS

Copies a double precision real array to a single precision real array.

COPYIN

Copies an integer array to another integer array.

COPYSD

Copies a single precision real array to a double precision real array.

CORDTR

Transforms the coordinate system defining the strain component from global to local, filters out specific components and transforms back to the original global coordinate system. Used by the strain-displacement matrix routines.

COROUT

Debug writes the nodal coordinates.

CPXBK1, CPXBK2

Back substitutes for the solution of complex matrix equations stored in the band-matrix form.

CPXDIV

Divides a complex number by another complex number.

CPXEXC

Sets up complex harmonic based excitation vector. The function is fully commented.

CPXFAC

Performs the crout decomposition of complex matrix equations stored in the banded form.

Common block: ALGEM

CPXFOR

Sets up the nodal force vector for the complex harmonic analysis.

CPXMUL

Multiplies two complex numbers.

CPXREA

Assembles the complex nodal reaction force vector due to the complex base excitation.

CPXRES

Calculates the harmonic nodal force.

CRPLAW

Defines the equivalent creep strain increment. To be used as a User Subroutine. See also the MHOST Version 4.2 Users' Manual.

CRPSTN

Determines quantities related to creep strain effects.

CUTHIL

Optimizes the band-width of global equations by Cuthil-McGee algorithm. Called by OPTIM.

D2DO4N

Calculates derivatives for two dimensional four node elements.

D2DO9N

Calculates derivatives for two dimensional nine node elements.

D3D08N

Calculates derivatives for three dimensional eight node hexahedral elements.

D3D27N

Calculates derivatives for three dimensional quadratic elements with 27 nodes.

DAMPIN

Reads in parameters defining damping terms in transient dynamic analysis.

Common block: MACHIN

DASHIN

Reads in the definition of additional damping terms in the form of discrete dashpots.

Common block: MACHIN

DAT1

Outputs an integer array on the line printer file.

Common block: ALGEM

DAT2

Outputs a real array on the line printer file.

Common block: ALGEM

DAT3, DAT5

Outputs a pair of integer and real arrays on the line printer file.

Common block: ALGEM

DATBMS

Prints out elastic beam section properties.

Common block: ALGEM

DATCG1

Modifies parameter data during a restart job.

Common block: ALGEM, CONTRO, ERRORS, HARMON, PARAM, PERIOD

DATER

Calls systems' routine to pull out today's date and time of the day. System dependent. See Section 0.4.

DATIN1

Supervises read operations of parameter data.

Common block: ALGEM, COUNT, CTITLE, ERRORS, FREE

DATIN2

Supervises read operations of model data and construction of in-core database.

Common block: ADDVAL, PULSES, POWER, EIGEN, MODSUP, HARMON, AUTOIN, BSECT, TIME, ALGEM, CONTRO, DAMP, MAXIMA, PARAM, START1, START2, START3, START4, START5, START7, SUBELM, SUBSTR, SHIFT, TOLER, INCCON, ERRORS, FREE

DATIN3

Supervises read operations of incremental data.

Common block: ALGEM, AUTOIN, CONTRO, COUNT, EIGEN, ERRORS, FREE, MAXIMA, PARAM, START1, START2, BODYFR, START3, START4, START5, START6, START7, TIME, TOLER, INCCON

DATOH1

Prints out model data as interpreted on the line printer.

Common block: ALGEM, CONTRO, PARAM, INCCON, BODYFR

DATOU1

Prints out parameter data as interpreted on the line printer.

Common block: ALGEM, COUNT, START4, EIGEN, HARMON, PULSES, CONTRO, TIME, AUTOIN, BSECT, MAXIMA, TMARCH, LOUBIN, PERIOD, SHIFT, PARAM, START1, START2, START3, START5, START7, SUBSTR, INCCON, TOLER, MACHIN

DATOU4

Prints incremental data as interpreted on the line printer.

Common block: ALGEM, CONTRO, PARAM, INCCON, TOLER

DAX04N

Calculates derivatives for four node axisymmetric elements.

DAX09N

Calculates derivatives for nine node axisymmetric elements.

DBM02N

Calculates derivatives for two node linear Timoshenko beam elements.

DECINT

Converts a given string to an integer.

Common block: ALGEM

DECOMP

Factorizes a symmetric matrix stored in profile form.

Common block: PARAM

DECREA

Converts a given string to a real.

Common block: ALGEM

DEFGUP

Updates the total deformation gradient at the end of an increment.

DERIV

Controls access to the entries in the element library.

Common block: ALGEM, CONTRO, ELTYP, ELEMEN, ERRORS

DIAM

Measures maxim bandwidth from a root node. Called by CUTHIL and used for bandwidth optimization.

Common block: ALGEM

DIRECT

Generates three orthogonal vectors, one of which is orthogonal to the plane defined by two given vectors.

Common block: ALGEM

DISPIN

Reads in initial displacements for transient analysis.

Common block: MACHIN

DISTIN

Reads in distributed load data.

Common block: MACHIN

DIV2Q2

Generates two dimensional subelement mesh composed of 9 node quadratic quadrilaterals.

DIV2X2

Generates two dimensional subelement mesh composed of 4 node linear quadrilaterals.

DMATIN

Reads in stress-strain matrix defined directly at nodes.

Common block: MACHIN

DMPING

Sets up the modal damping matrix for the current element.

DOT (Function Subprogram)

Calculates dot-product of two double precision real vectors.

DSH04N

Calculates derivatives for four node shell elements in the lamina coordinate system.

DSHELL

Converts the notation between tensorial and matrix forms for stress-strain matrix.

DSLOAD

Controls the access to the element distributed load routines.

Common block: CONTRO, ELTYP, ELEMEN, MAXIMA, PARAM, START1, START2, START3, START4, START5, START6, START7, MACHIN

DUPLIN

Reads in duplicate node definition data.

Common block: MACHIN

DXOUT

Prints out a double precision array for debugging purposes.

DYNAMT

Drives the transient time integration for dynamic analysis.

Common block: ALGEM, AUTOIN, PERIOD, CTITLE, DAMP, EIGEN, INCCON, ELEMEN, ELTYP, MAXIMA, CONTRO, SUBSTR, ERRORS, PARAM, START1, START2, MACHIN, START3, TMARCH, START4, START5, START6, TIME

DYNOP

Assembles the element operator matrix for transient time integration.

Common block: ALGEM, TMARCH

EIGENV

Drives the eigenvalue extraction subsystem by subspace iteration.

Common block: CONTRO, ELEMEN, EIGEN, SHIFT, ELTYP, PARAM, MACHIN, START3, START4, START5, START7

ELVULV

Pulls out parameters defining element characteristics from the workspace.

Common block: ALGEM, CONTRO, ELTYP, START1, PARAM, ELEMEN

EQVC

Calculates the strain invariant from the given deviatoric strain tensor stored in vector form.

EQVS, EQVSTR

Calculates the stress invariant from the given deviatoric stress tensor stored in vector form.

ERROR

Prints error message on the line printer output file.

Common block: ALGEM

ETRANS

Transforms coordinates for the nodal displacement vector according

to user instructions.

Common block: ALGEM

FILINT

Fills an integer array with its address or its inverse order.

Common block: ALGEM

FILL

Fills a double precision real array with a specified number.

FIRST1

Re-numbers the nodes for Cuthill-McKee optimization.

FIXINT

Imposes constraints onto the frontal solution.

External reference: FILINT, VULVRG, COPY

FOLOIN

Reads parameters specifying the concentrated nodal follower force.

Common block: MACHIN

FORRES

Calculates forced vibration response by modal superposition.

FREDOM

Drives the linear dynamic analysis in frequency domain.

Common block: CTITLE, TIME, AUTOIN, START1, START2, START3, START4, START5, START7, START8, ALGEM, EIGEN, MODSUP, POWER, DAMP, INCCON, MACHIN, ELEMEN, MAXIMA, ELTYP, CONTRO, SUBSTR, ERRORS, PARAM, SHIFT

External reference: STRUCT, TIMOUT, ASSEM1, SOLUT1, ASSEM4, RESCON, ADDINC, PRINOU, POSTOU, COPY, NUL, INITST, SUBT, MASMAT, MADD, EIGENV, CNSTNM, PNTTNM, INTSQQ, SETRMS, QUIT

FREFOR

Reads free format numeric data line from the card reader.

Common block: ALGEM, FREE

FRNTBL

Sets up the elimination table for the frontal solution and calculates maximum front matrix size.

Common block: ALGEM, PARAM, MAXIMA, MACHIN

FRNTOP

Optimizes element numbering for the frontal solution. Not fully functional in Version 4.2.

FRONTB

Performs back substitution of global equations factorized the by the frontal process.

FRONTF

Performs forward elimination of global equations by the frontal process.

FRONTR

Performs resolution by frontal solution for a newly defined load vector with a readily factorized system of equations.

FRONTS

Drives the quasi-static solution process utilizing the frontal solution scheme.

Common block: CTITLE, TIME, AUTOIN, START1, START2, START4, START5, START6, ALGEM, EIGEN, INCCON, MACHIN, ELEMEN, MAXIMA, ELTYP, CONTRO, SUBSTR, ERRORS, PARAM, START8

FRONTW

Computes the maximum front width from the element connectivity table.

Common block: ALGEM, MAXIMA, MACHIN

GAUSSP

Returns coordinates and weight factors for Gaussian quadrature.

GBM02N

Calculates derivatives for two node linear Timoshenko beam element with respect to element coordinate system.

GENCOR

Generates coordinates for subelement nodes which divide global

element uniformly.

GENNOD

Generates connectivity table for uniformly divided subelement mesh.

GEOMAT

Assembles the geometric stiffness for large displacement and buckling analyses.

External reference: GAUSSP, COPY, S2D04N, S3D08N, D2D04N, DAX04N, DSH04N, D3D08N, NOTION

GETBSP

Extracts data for beam section properties.

Common block: ALGEM

G02GLO

Rotates stresses and strains back to global configuration in finite deformation analysis.

External reference: ROTTEN, ROTBAK

G02ROT

Rotates stresses and strains to mid-increment configuration in finite deformation analysis.

External reference: ROTTEN, ROTFOR

GRAV2D

Produces body force vector due to gravity acceleration for two-dimensional elements.

Common block: ALGEM

GRAV3D

Produces the body force vector due to gravity acceleration for three-dimensional elements.

GRAVAX

Produces the body force vector due to gravity acceleration for axisymmetric elements.

GRAVBM

Produces the body force vector due to gravity acceleration for beam elements.

GRAVSH

Produces the body force vector due to gravity acceleration for shell elements.

GSH04N

Calculates the element normal to four node shells.

HARMIN

Reads in parameters for harmonic base excitation.

Common block: ALGEM

HEAD

Prints the MHOST banner and system's information on the line printer output file and terminal screen.

Common block: CTITLE

HOLECR

Generates coordinate data for an embedded hole represented by a subelement mesh. Called by HOLEDF.

HOLEDF

Defines an element with a parameterized hole using the subelement procedure.

Common block: MAXIMA, ELTYP, SUBTYP, MACHIN, START1, TRANSF, START6

External reference: ELVULV, QNODEL, HOLECR, HOLELM

HOLEIN

Reads parameters to define a embedded hole in a global element.

Common block: ELTYP, SUBELM, MACHIN

HOLELM

Generates connectivity array for an embedded hole represented by a subelement mesh. Called by HOLEDF.

HOOKBM

Calculates the linear elastic stress-strain matrix for beam elements.

External reference: NUL, GETBSP, MULT

HOOKLW

Calculates the linear elastic stress-strain matrix for all element types other than beams.

External reference: NUL, UTEMP, UHOOK

HOST

Supervises execution of the MHOST code. See Section 1.2 for detail.

Common block: CONTRO, ALGEM, ADDVAL, TMARCH, AUTOIN, PERIOD, POWER, PULSES, EIGEN, ERRORS, MODSUP, HARMON, LOUBIN, START5, START4, PARAM, BSECT, INCCON, MACHIN, SUBSTR, SUBELM, SHIFT

ICLEAR

Zero clears an integer array.

INCRIN

Drives the reader and report writer for incremental data.

External reference: DATIN3, DATOU4, DATOU1

INCSBC

Imposes stress boundary conditions.

INIMOP

Creates initial eigenvectors for subspace iteration.

Common block: CONTRO, MAXIMA, ELTYP, ELEMEN, PARAM, EIGEN, START1, START2, START3, START4, START5, START6, START7, START8, MACHIN

External reference: QUIT, NUL, COPY, RATIO, MAXIM

INITDF

Initializes the nodal deformation gradient array.

INITFR

Allocates memory for frontal solution option.

Common block: ALGEM, CONTRO, PARAM, MACHIN, MAXIMA, EIGEN, START5, START6

INITI1

Allocates memory for in-core nodal database.

Common block: ADDVAL, ALGEM, CONTRO, DAMP, EIGEN, MODSUP, HARMON, POWER, PULSES, MAXIMA, PARAM, TMARCH, PERIOD, SUBELM, SHIFT, SUBTYP, BSECT, START1, START2, START3, START4, START5, START6, START7, START8, START9, SUBSTR, MACHIN, ERRORS

INITI2

Allocates memory for the profile solution for quasi-static, dynamic and eigenvalue analyses.

Common block: ALGEM, CONTRO, PERIOD, EIGEN, ERRORS, MAXIMA, PARAM, START1, START3, START5, START4, MACHIN

INITIN

Reads parameters defining initial conditions for transient analysis.

Common block: ALGEM, ERRORS, CONTRO, MAXIMA, ELEMEN, ETYPE, START1, START5, PARAM, FREE, START2, START4

External reference: KEY, DISPIN, VELCIN, ACCLIN, PERDIN, QUIT

INITSE

Allocates memory for the subelement nodal database. See Section 2.2 for detail.

Common block: SUBTYP, ALGEM

External reference: SUBELV, NULINT

INITST

Assembles the global initial stress matrix for quasi-static, dynamic and eigenvalue analysis.

Common block: AUTOIN, EIGEN, CONTRO, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, SUBSTR, TIME, TRANSF, MACHIN

External reference: NUL, ELVULV, CNODEL, GEOMAT, ASSEM5

INRDIR

Redirects the input stream.

Common block: ALGEM, FREE

INRFRC

Recovers the velocity and acceleration from the displacement update in transient dynamics.

Common block: TMARCH

INSIDE

Checks to see if the element is inside-out and repairs the connectivity array, if necessary, for the linear quadrilateral and hexahedral elements.

INTDYN

Initializes the scratch file for transient dynamics.

INTERP

Interpolates nodal values to the integration points.

INTINT

Initializes the file system and execution environment. System dependent. See Section 0.4. Called by MAIN program.

Common block: VRTDSK

INTSQQ

Integrates the spectral density function over a given range of frequencies.

INV3

Inverts a 3 by 3 matrix.

INVERT

Inverts a general square matrix.

Common block: ALGEM

ITERIN

Reads parameters defining convergence criteria.

Common block: TOLER

External reference: FREFOR

JACOB1

Extracts eigenvalues of a square matrix. Called by SUBSPC.

Common block: ALGEM

External reference: MAXIM, SMULT, LINES

JT (Function subprogram)

Gives the address of a specified entry in a global array stored in band matrix form.

KEY

Checks to see if a string matches to a keyword.

Common block: ALGEM, FREE

L2NORM

Calculates 1 norm of a given vector.

LAXSYM

Calculates the load vector due to body force loading on axisymmetric elements.

Common block: ALGEM

LAYINT

Integrates lamina quantities to resultants for thick shell elements.

LELAST

Calculates linear elastic response of a material for a pre-defined stress-strain matrix.

LETCMD

Sets up the integer control variables for the compound execution of analysis drivers. See Section 1.2 for detail.

Common blocks: ALGEM, CONTRO, COMPND, PARAM, EIGEN, MACHIN

LEVEL

A part of the front matrix size optimization package. Not used in Version 4.2.

LINES

Advances the line number of the line printer output file (ILPRNT).

Common block: ALGEM

External reference: PAGE

LINE52

Advances the line number of the line printer output file (JLPRNT).

Common block: MAXIMA, ALGEM, ZPRINT

External reference: PAGE2, PAGE3

LINESR

Calculates search distances for the line search algorithm.

Common block: MACHIN, CONTRO, ALGEM, AUTOIN, PARAM, START3, START4, START5, START7, TOLER

LINESU

Advances the line number of the line printer output page used for the subelement solution.

Common block: MAXIMA, ALGEM, ZPRINT

External reference: PAGE2S, PAGE 3S

LMPMAS

Calculates the diagonalized Gramm matrix for nodal strain projections. Either Gauss-Lobatto or Gauss/now-sum algorithm can be used.

LOCVEC

Fills up the address table for the global array stored in band matrix form.

Common block: ELTYP

LPSTRN

Calculates the load vector due to body force loading on plane strain elements.

LPSTRS

Calculates the load vector due to body force loading on plane stress elements.

LSHELL

Calculates the load vector due to body force loading on shell elements.

LSOLID

C - 2

Calculates the load vector due to body force loading on solid elements.

LTBEAM

Calculates the load vector due to body force loading on beam elements.

MADD

Adds a matrix to a product of another matrix and a scalar.

MAIN

Main program.

Common block: blank, MACHIN

MASMAT

Assembles the global consistent matrix stored in profile form.

Common block: AUTOIN, EIGEN, CONTRO, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, BSECT, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, SUBSTR, TIME, TRANSF, MACHIN

MASSIN

Reads in the connectivity and magnitude of added mass.

Common block: MACHIN

MATINV

Inverts a given square matrix.

MATONE

Fills the diagonal entry of a global matrix by unity.

MATPRT

Prints out a two dimensional array with a header.

Common block: ALGEM

MATSUM

Adds two arrays multiplied by given constants for each array.

MAXCON

Finds the maximum connectivity at nodes used for bandwidth optimization.

MAXIM

Pulls out the maximum value from a double precision array.

PERDIN

Reads data defining the periodic loading in transient dynamics.

Common block: MACHIN, ELTYP, PERIOD

PERDOP

Applies periodic loading to the current time integration step.

Common block: ALGEM

PJOOP

Debug - writes a double precision real variable/array.

PLASTD

Calculates a consistent tangent modulus for an elastic plastic material at a given state of stress and deformation history.

Common block: ELEMEN

PLASTS

Calculates the stress state of an elastic-plastic material by return mapping.

Common block: ELEMEN

PAGE

Advances a page of line-printer output file (ILPRNT) and prints the header.

Common block: ALGEM, CTITLE, PAGCNT

PAGE2

Advances a page of line-printer output file (JLPRNT) and prints the header for nodal variables.

Common block: ALGEM, CONTRO, CTITLE, RESULT, TIME, PAGCNT

PAGE3

Advances a page of line-printer output file (JLPRNT) and prints the header for element variables.

Common block: ALGEM, CONTRO, CTITLE, RESULT, TIME, PAGCNT

PAGE25

Same as PAGE2. Used for subelement solution.

PAGE35

Same as PAGE3. Used for subelement solution.

NUL

Zero - clears a double precision real array.

NULINT

Zero - clears an integer array.

NULNRM

Zero - clears space allocated for nodal normals of shell coordinate definition.

OPTIM

Controls execution of the bandwidth optimization option.

External reference: LINES, MKFAKE, MAXCON, CUTHIL, QUIT

ORIENT

Generates the coordinate transformation matrix to the preferred orientation of an anisotropic material.

OUTPRO

Performs the outer product of two three-dimensional vectors.

Common block: START2, START3, START4, START5, START6, START7, START8, SUBSTR, TIME, TRANSF, MACHIN

NEWVEL

Updates the velocity vector at the beginning of a time integration step.

NODPRE

Computes the nodal pressure and directly adds it into the global load vector.

NODSTR

Computes stress at a given point and generates material tangent.

External reference: NUL, COPY, ROTPRF, THRSTN, CRPSTN, SUBT, SIMPLE, MULT, LELAST, PLASTS, PLASTD, WALKEQ

NOTION

Converts stresses and strains stored in vector form to tensorial form.

NRMNRM

Normalizes the nodal normal vector and stores its components as a part of the coordinate data for shell elements.

MULT

Multiplies two two-dimensional matrices.

MULTT

Multiplies a matrix by the transpose of another matrix.

NEWACC

Updates the acceleration vector at the beginning of a time integration step.

NGWADD

Allocates the acceleration and velocity vector at the beginning of a time step to temporary storage.

NEWMRK

Adds the element stiffness and mass matrices to generate the element time integration operator matrix.

External reference: MADD

NEWRHS

Forms global load vector.

Common block: AUTOIN, BSECT, CONTRO, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, PULSES, MAXIMA, PARAM, START1

MESURE

Calculates a distance from the integration point to a straight line defined by two points.

MID

Calculates the coordinate of the mid edge node. Not used in Version 4.2.

MIDDLE

Drives the calculation of coordinates and connectivity for mid edge nodes. Not used in Version 4.2.

MKFAKE

Sets up a fake connectivity table including tying constraint for bandwidth optimization.

MODAL

Drives the modal analysis.

Common block: CTITLE, TIME, AUTOIN, START1, START4, START5, ALGEM, EIGEN, INCCON, MACHIN, ELEMEN, MAXIMA, ELTYP, CONTRO, SUBSTR, ERRORS, PARAM, SHIFT, START8

External reference: STRUCT, TIMOUT, ASSEM1, MASMAT, MADD, EIGENV, POSTOU

PNTTNM

Sets up the convolution terms for nodal loads.

POLD2D

Performs polar decomposition of a two-dimensional second order tensor.

POLD3D

Performs polar decomposition of a three-dimensional second order tensor.

POLICE

Debug - writes contents of a specified common block.

POSTEN

Packs the record of nodally defined element variables for the post tape. See Section 3.4 for detail.

Common block: POSTPN

POSTOU

Writes out post processing data. See Section 3.4 for detail.

POSTPR

Writes contents of buffer to the post processing tape.

PREFIN

Reads in the parameters defining the preferred orientation of an anisotropic material.

Common block: MACHIN

PRELEM

Extracts information of the current element from the nodal database. A simplified version of CNODEL used in PRINOU.

Common block: CONTRO, ELTYP, MAXIMA, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, MACHIN

PRESET

Initializes working arrays used for front matrix size optimization.

Common block: ALGEM

PRESIN

Reads parameters defining pressure loads.

Common block: MACHIN

PRFRNT

Controls the front matrix size calculation and the subsequent memory allocation.

Common block: CONTRO, PARAM, ELEMEN, ELTYP, MAXIMA, MACHIN, START1, START2, START3

PRINCV

Solves a 3 by 3 eigen problem by Jacobi iteration.

PRININ

Reads control data for line printer output.

Common block: FREE

PRIN01, PRIN02, PRIN03

Writes an array with a header.

PRINOU

Controls the line printer output of the global solution.

Common block: CONTRO, ALGEM, ELEMEN, ELTYP, RESULT, START1, START2, START4, START6, SUBELM, TIME, TRANSF, ZPRINT, MACHIN

PRINSU

Controls the line printer output of the subelement solution.

Common block: CONTRO, ALGEM, ELEMEN, ELTYP, RESULT, START1, START2, START4, START6, TIME, TRANSF, ZPRINT, MACHIN

PRINTM

Debug - writes the element stiffness matrix.

Common block: ALGEM, ELEMEN

PRINTS

Debug - writes the global stiffness matrix stored in band matrix form.

Common block: ALGEM

PRNSHL

Prints the nodal coordinate transformation matrix for shells.

PRNTEL

Reports the connectivity table resulting from automatic subelement mesh generation.

PRNTNO

Reports the coordinates resulting from automatic subelement mesh generation.

PRTERR

Prints error message and does not STOP but returns the control.

Common block: ALGEM

PRWARN

Prints warning message.

Common block: ALGEM

PSDIN

Reads the power spectrum definition.

Common block: MACHIN

PULSIN

Reads the pulse load definition.

Common block: MACHIN

PUTDUP

Adds the duplicate node option as a special element in the frontal solution subsystem.

PUTTIE

Adds the tying equation as a special element in the frontal solution subsystem.

QNODEL

Extracts element quantities from the global nodal database. This is yet another subset of CNODEL.

Common block: TMARCH, ALGEM, CONTRO, ELTYP, MAXIMA, PARAM, START1, START2, START4, START5, START6, START7, START8, START9, MACHIN

QUIT

Prints the error message and terminates the execution.

Common block: ALGEM

R3DTEN

Defines three-dimensional rotation matrix for decomposition of the deformation gradient.

RAMDSK

Initializes the buffer area used for the out-of-core frontal solution.

Common block: VRTDSK, ALGEM, PARAM, MACHIN, START5

RATIO

Subdivides entries of one array by corresponding entries of another array.

RBF

Rotates stress and strain vectors to preferred orientations.

READEX

Reads header and a double precision real array from a binary file.

REASPR

Calculates reaction force due to an added spring.

REDIAG

Reduces the diagonal entries of a profile store global stiffness

matrix.

RELDFG

Computes the nodal incremental deformation gradient.

RELOAD

Forms the follower force at the end of an increment.

Common block: ALGEM, ADDVAL, AUTOIN, BSECT, CONTRO, BODYFR, LOUBIN, ELEMEN, ELTYP, ERRORS, MAXIMA, PARAM, PULSES, START1, START2, START3, START4, START5, START6, START7, START8, SUBELM, SUBSTR, TIME, TRANSF, MACHIN

External reference: NUL, ADD, RTFOLF, ELVULV, CNODEL, DERIV, INTERQ, STIFF, ETRANS, SUBST1, BFLOAD, DSLOAD, QUIT

RESCHK

Tests for convergence flag associated with the residual.

RESCON

Tests for convergence in terms of residual reaction and displacement.

Common block: ALGEM, CONTRO, PARAM, START1, START2, START3, START4, START5, START6, START7, START8, SUBELM, SUBSTR, TIME, TRANSF, MACHIN

External reference: ADD, SUBT, TYING2, BOUND2, MAXIM, L2NORM, LINES, SUBT

RESDYN

Calculates the contribution of mass and damping terms to the residual vector. Called by ASSEM4 when the direct time integration flag is on.

Common block: TMARCH, ALGEM

RESEQ1

Renumbers nodes for the reduction of front width. Not functional in Version 4.2.

RESEQ2

Renumbers elements for the reduction of front width. Not functional in Version 4.2.

RESID

Forms the residual vector. Called by ASSEM4.

Common block: PARAM

RESOLV

Resolves the same equation by frontal solution method for a different load vector.

Common block: ALGEM

RESTRT

Reads the restart file. See Section 3.3 for detail.

Common block: ADDVAL, ALGEM, AUTOIN, BODYFR, BSECT, CONTRO, COMPND, COUNT, DAMP, EIGEN, MODSUP, ELEMEN, HARMON, INCCON, MAXIMA, LOUBIN, PARAM, PERIOD, POSTPN, POWER, PULSES, RESULT, START1, START2, START3, START4, START5, START6, START7, START8, START9, SUBELM, SUBTYP, SUBSTR, SHIFT, TIME, TMARCH, TOLER, VRTDSK, ZPRINT

ROTBAK

Rotates the stress state back to the global coordinate system in finite deformation analysis.

ROTDMT

Rotates the element stress strain matrix into the global coordinate system for shells.

External reference: ORIENT, COPY, QUIT, MATINV, NUL, TSHIFT, TFULL2, MULT

ROTFOR

Rotates the stress state state into the deformed configuration coordinate system in finite deformation analysis.

ROTPRF

Rotates the stresses and strains defined in global coordinates to the preferred orientation of an anisotropic material.

ROTTEN

Calculates the rotation tensor from the polar decomposition of the deformation gradient in finite deformation analyses.

ROW

Pulls out a complete row from profile stored global matrix.

RSHELL

Treats the random vibration input for shell elements.

RTBEAM

Treats the random vibration input for beam elements.

RTFOLF

Rotates load vector entries treated as follower forces.

RUNCMD

Loads control parameters for compound executions.

Common block: CONTRO, COMPND, MACHIN

SID02N

Shape functions for the one-dimensional two node element.

S2D04N

Shape functions for the two-dimensional four node element.

S2D09N

Shape functions for the two-dimensional nine node element.

S3D08N

Shape functions for the three-dimensional eight node element.

S3D27N

Shape functions for three-dimensional twenty-seven node element.

SAVER:

Writes the restart file. See Section 3.3 for detail.

Common block: ADDVAL, ALGEM, AUTOIN, BODYFR, BSECT, CONTRO, COMPND, COUNT, DAMP, EIGEN, MODSUP, ELEMEN, HARMON, INCCON, MAXIMA, LOUBIN, PARAM, PERIOD, POSTPN, POWER, PULSES, RESULT, START1, START2, START3, START4, START5, START6, START7, START8, START9, SUBELM, SUBTYP, SUBSTR, SHIFT, TIME, TMARCH, TOLER, VRTDSK, ZPRINT

SBCIN

Reads in stress boundary condition data.

Common block: MACHIN

SCALER

Scales loading to a given proportion.

Common block: MACHIN, START8

SEARCH

Pulls out double precision real element nodal quantities from the global nodal data base.

SEARCI

Pulls out integer element nodal quantities from the global nodal database.

SELECT

Reads and decodes strings for print option data input.

SETCCM

Sets up the complex coefficient matrix for base excitation.

SETHFN

Sets up the vector for complex modal damping.

SETOLR

Sets the limit for the search distance in the line search option.

SETOMD

Calculates modal damping.

SETRMS

Finds the root-mean-square value for a modal function from the frequency response.

SETUP

Computes the adjacency list for front width optimization. Not functional for Version 4.2.

SHIFIN

Reads parameters defining the power shift in modal dynamics.

Common block: MACHIN

SHOHEI

Debug - writes an integer array.

SHTRAN

Transforms nodally defined shell stress and strain resultants to

values in element coordinates.

SIMPLE

Calculates the stress and generates the tangent modulus for secant elasticity model for elastoplastic response under a monotonically increasing load.

SK

Integrates the element coefficient matrix into the global array stored in band matrix form.

SMASTR

Extracts element quantities from the global element used for subelement analysis. Yet another variation of CNODEL.

Common block: TMARCH, CONTRO, ELTYP, MAXIMA, PARAM, START1, START2, START4, START5, START6, START8, MACHIN

SMULT

Multiplies an array by a scalar.

SNODEL

Transfers global element results into the subelement mesh.

SOLUT1

Controls execution of the linear algebraic equation solver by the profile method.

Common block: AUTOIN, CONTRO, EIGEN, ELTYP, MAXIMA, PARAM, PULSES, START1, START3, START4, START5, START7, MACHIN, TOLER

External reference: ADD, RTFOLF, ADDPUL, SUBT, TYING1, BOUND1, DECOMP, BFGSVW, BFGSRH, COPY, SOLVER, MADD, MATSUM, BFGSLH, TYING3, L2NORM, TYING2, BOUND2, ADAPTS

SOLUT2

Controls execution of the linear algebraic equation solver by the band matrix method. Not used in Version 4.2.

SOLVER

Solves linear equations for nodal displacement.

Common block: ALGEM, PARAM

SOLVIT

Solves a small system of algebraic equations by Gaussian elimination.

SPRIIN

Reads parameters defining the added spring stiffness.

Common block: MACHIN

SPSTRS

Generates the strain displacement matrix at a given point in the current element of plane stress type.

SSOLID

Generates the strain displacement matrix at a given point in the current element of three-dimensional continua type.

STATIC

Drives the incremental iterative solution of a quasi-static problem.

STCKIO

Controls the data flow between in-core buffer and the work file for the frontal solution.

Common block: ALGEM, MACHIN, VRTDSK

STIFF

Calculates the current element tangent stiffness matrix and the load vector.

External reference: NUL, MATPRT, SMULT, MULT, TMULT

STRAIN

Recovers strain at nodes.

Common block: COUNTR

STRESS

Controls the nodal stress recovery operation including pre-integration through the shell thickness.

STRING

Decodes the input data string.

Common block: ALGEM, FREE, COUNT

STRIPB

Finds non-blank entry in a given character string.

STRSBC

Imposes the stress boundary condition called by ASSEM4.

STRUCT

Controls the core allocation and elimination table construction for global matrix manipulation.

Common block: ALGEM, CONTRO, MAXIMA, PARAM, START1, START2, START3, START4, START5, SUBSTR

SUBALC

Allocates working storage for the subelement solution.

SUBCHK

Checks convergence of the subelement solution in terms of displacement update.

Common block: ALGEM, SUBCNV

SUBDER

Controls access to the element library in the subelement solution.

Common blocks: ALGEM, CONTRO, SUBTYP, ELEMEN, ERRORS

SUBDIV

Controls the subelement mesh generation.

Common blocks: ALGEM, CONTRO, ELTYP, ERRORS, MACHIN, MAXIMA, PARAM, START1, START5, START6, SUBTYP

SUBELV

Extracts element definition variable for the current subelement.

Common block: ALGEM, SUBTYP, START1, PARAM, ELEMEN

SUBFEM

Supervises the mixed iterative solution in the subelement mesh. Called by ASSEM4.

Common block: ALGEM, CONTRO, ELTYP, ELEMEN, SUBTYP, TRANSF, LOUBIN,

MAXIMA, PARAM, ERRORS, MACHIN, TIME, START1, START2, START3, START4, START5, START6, START7, START8, START9, TOLER

External reference: TIMOUT, SUBELV, SUBALC, SMASTR, NUL, SNODEL, LMPMAS, SUBINT, SUBT, STRESS, ADD, SUBDER, INTERP, STIFF, SYSEQN, SUBSOL, STRAIN, SUBRES, SUBCHK, SUBGLB, SUBGLD

SUBGLB, SUBGLD

Calculates residual for a global element generated by the subelement solution.

SUBINC

Adds the incremental values of the subelement solution to the total array.

Common block: AUTOIN, CONTRO, SUBTYP, START1, START2, START4, START6, START8, TIME

SUBINT

Interpolates global quantities at subelement nodes.

SUBRES

Calculates the mixed residual for the subelement solution.

SUBSIN

Reads in substructure data. Not used in Version 4.2.

SUBS01, SUBS02

Back - substitutes for the factors of the global array stored in band matrix form to generate the displacement corresponding to the given load vector. Not used in Version 4.2.

SUBSOL

Solves the subelement stiffness equations for the subelement displacement.

SUBSPC

Controls execution of the subspace iteration for eigenvalue extraction.

Common block: ALGEM, SHIFT

External reference: NUL, UPTX, JACOBI, TIMOUT, COPY, MADD, QUIT, SMULT, LINES

SUBST1

Converts the element load vector to the global load vector.

SUBSTN

Adds the element nodal value to the global vector.

SUBT

Subtracts arrays.

SUBVAL

Interpolates element nodal values to a given point specified by the isoparametric coordinate system.

SUPER

Drives modal superposition for linear dynamics.

Common block: CTITLE, TIME, AUTOIN, START1, START2, START4, START5, ALGEM, EIGEN, MODSUP, HARMON, INCCON, DAMP, MACHIN, ELEMEN, MAXIMA, ELTYP, CONTRO, SUBSTR, ERRORS, PARAM, SHIFT, START8

SYSEQN

Assembles the global finite element equations for subelement analysis.

T2D04N

Calculates the local coordinate system for two-dimensional four node elements.

T2D04P

Calculates the local coordinate system for two-dimensional four node elements using the Cayley-Hamilton formula.

T3D08N

Calculates the local coordinate system for three-dimensional eight node element.

TBM02N

Calculates the local coordinate system for two node beam element.

TEMPIN

Reads in nodal temperature definition data.

Common block: MACHIN

TFULL2

Transforms the material modulus into fully three-dimensional form.

THRSTN

Calculates thermal strain.

External reference: UTEMP, NUL, UCOEF

TIMEIN

Reads in parameters defining the time increment control.

TIMER

Accesses to the system dependent CPU clock routine. See Section 0.3.

TIMOUT

Reports the job step and elapsed CPU time.

Common block: ALGEM, CONTRO, TIMLOC

TMULT

Multiplies the transpose of an array by a matrix.

TMULTV

Multiplies the transpose of an array by a vector.

TNSPRD

Multiplies two tensors stored in vector form.

TRACIN

Reads in nodal concentrated forces.

Common block: MACHIN

TRANIN

Reads nodal coordinate transformation data.

Common block: MACHIN

TRANS1

Applies a transformation to the global stiffness matrix stored in band matrix form.

TRANS2

Applies a transformation to the global vector.

TRANSP

Transposes a two-dimensional matrix.

TSH04N

Calculates local coordinates for four node shell elements.

TSHIFT

Sets up tensor transformation with respect to the preferred orientation of an anisotropic material.

TYING1

Imposes tying constraints on the global stiffness matrix stored in band matrix form.

TYING2

Imposes tying constraints on the global vector.

TYING3

Releases tying constraints in the global vector.

TYININ

Reads in coefficients of the tying equations.

Common block: MACHIN

TYPEIN

Defines element parameters used in the analysis.

UBOUN, UCOEF, UCOOR, UDERIV, UFORCE, UHOOK, UPRESS, UTEMD, UTHERM, USXX, VSWELL

User subroutines. Fully documented in MHOST Volume 1 USERS' MANUAL.

UNITST

Generates the global unit matrix in profile stored form.

UPTX

Multiplies a banded matrix by a vector.

UPTXL

Multiplies a lumped mass matrix by a vector.

VALINT

Linearly interpolates data given in a table.

VDSKID

Manages a record for out-of-core frontal solution.

Common block: ALGEM, MAXIMA, MACHIN, VRTDSK

VELCIN

Reads initial velocity definition data.

Common block: MACHIN

VMULT

Multiplies an array by a vector.

VSH04N

Generates the local coordinate system for four node shell elements.

VTMULT

Multiplies the transpose of an array by a vector.

VULVRG

Makes the conversion table for the degree of freedom from the connectivity and constraint data.

VVMULT

Multiplies two vectors.

WALCON

Calculates temperature dependent material constraints for the Walker model.

WALKEQ

Calculates stress and internal variables for the Walker model.

WORKIN

Reads in work-hardening data for a homogeneous elastic plastic material.

Common block: MACHIN

WKSLP

Calculates the hardening slope for elastoplasticity.

WRITEX

Writes a record on a binary file.

YIEL

Calculates the yield stress and hardening slope for a given equivalent plastic strain.

External reference: WKSLP

YIELIN

Reads in the nodal definition of a stress-strain curve for an elastic plastic material.

Common block: MACHIN